

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО”**

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

**До захисту допущено:  
В. о. завідувача кафедри  
Михайло НОВОТАРСЬКИЙ**

\_\_\_\_\_ (підпис)

“\_\_” \_\_\_\_\_ 2025 р.

**Дипломний проєкт**

**на здобуття ступеня бакалавра**

**за освітньо-професійною програмою “Комп’ютерні системи та мережі”**

**спеціальності 123 “Комп’ютерна інженерія”**

на тему: Веб-застосунок для економічного аналізу азійського регіону

Виконав: студент 4 курсу, групи Ю-15  
(шифр групи)

Гонтар Данііл Сергійович

(прізвище, ім’я, по батькові)

\_\_\_\_\_ (підпис)

Керівник асистент, Кулаков Ю. О.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

\_\_\_\_\_ (підпис)

Консультант(нормоконтроль) асистент, Нікольський С.С.

(назва розділу)

(посада, вченезвання, науковийступінь, прізвище та ініціали)

\_\_\_\_\_ (підпис)

Рецензент доцент, Шимкович В. М.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

\_\_\_\_\_ (підпис)

Засвідчую, що у цьому дипломному  
проєкті немає запозичень з праць інших  
авторів без відповідних посилань.

Студент \_\_\_\_\_

(підпис)

Київ – 2025 р.

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО”**

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Рівень вищої освіти – перший (бакалавр)

Освітньо-професійна програма

“Комп’ютерні системи та мережі”

спеціальності 123 – Комп’ютерна інженерія

**ЗАТВЕРДЖУЮ**

**В. о. завідувача кафедри**  
**Михайло НОВОТАРСЬКИЙ**

\_\_\_\_\_ (підпис)

“ \_\_\_ ” \_\_\_\_\_ 2025 р.

**ЗАВДАННЯ**

на бакалаврський дипломний проєкт студента

Гонтаря Данііла Сергійовича

1. Тема проєкту Веб-застосунок для економічного аналізу азійського регіону  
керівник проєкту Кулаков Юрій Олексійович, асистент  
(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)  
затверджені наказом по університету від 23 травня 2025 року № 1705-с.

2. Термін здачі студентом закінченого проєкту 2 червня 2025 року

3. Вихідні дані до проєкту див. технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які розробляються)

Розділ 1. Огляд існуючих рішень для економічного аналізу

Розділ 2. Огляд та вибір технологій для розробки системи

Розділ 3. Деталі розробки системи

Розділ 4. Дослідження та аналіз розробленої системи

5. Перелік графічного матеріалу (з точним позначенням обов'язкових креслень) алгоритм дії розробленої системи, діаграма класів, структурна модель системи.

6. Консультанта проєкту, з вказівкою розділів проєкту, які до них вносяться

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Нормоконтроль	Нікольський С.С		

7. Дата видачі завдання 08 лютого 2025 року.

#### Календарний план

№ П/П	Найменування етапів дипломного проєкту	Терміни виконання етапів проєкту	Примітки
1.	<i>Затвердження теми роботи</i>	<i>28.01.2025</i>	
2.	<i>Вивчення та аналіз завдання</i>	<i>01.02.2025 - 11.04.2025</i>	
3.	<i>Розробка архітектури та загальної структури системи</i>	<i>25.04.2025 - 01.05.2025</i>	
4.	<i>Розробка структур окремих частин системи</i>	<i>01.05.2025 - 15.05.2025</i>	
5.	<i>Програмна реалізація системи</i>	<i>15.05.2025 - 29.05.2025</i>	
6.	<i>Оформлення пояснювальної записки</i>	<i>29.05.2025 - 02.06.2025</i>	
7.	<i>Захист програмного продукту</i>	<i>02.06.2025</i>	
8.	<i>Передзахист</i>	<i>10.06.2025</i>	
9.	<i>Захист</i>	<i>18.06.2025</i>	

Студент-дипломник \_\_\_\_\_ **Данііл ГОНТАР**  
(підпис)

Керівник проєкту \_\_\_\_\_ **Олексій КУЛАКОВ**  
(підпис)

## АНОТАЦІЯ

У цій бакалаврській роботі розроблено вебзастосунок для економічного аналізу азійського регіону, що надає можливість інтерактивного аналізу та візуалізації даних провідних компаній за секторами та країнами. Система побудована за архітектурою клієнт-сервер, де серверна частина використовує Node.js і Express для обробки фінансових даних з Yahoo Finance API, а клієнтська реалізована на React для зручного користувацького інтерфейсу та ефективної взаємодії. Застосунок підтримує порівняння економічних показників, відстеження ринкової динаміки та аналітичні дослідження, що забезпечує комплексне розуміння економічних тенденцій Азії.

Ключові слова: економічний аналіз, вебзастосунок, Азія, React, Node.js, клієнт-сервер, Yahoo Finance API, візуалізація даних, фінансові дані, Express, інтерфейс користувача.

## ANNOTATION

This bachelor's thesis presents a web application for economic analysis of the Asian region, enabling interactive analysis and visualization of data on leading companies categorized by sectors and countries. The system follows a client-server architecture, with the server side implemented using Node.js and Express to process financial data from the Yahoo Finance API, and the client side built with React for an intuitive user interface and efficient interaction. The application supports comparative analysis of economic indicators, market dynamics tracking, and analytical research, providing comprehensive insights into the economic trends of Asia.

Keywords: economic analysis, web application, Asia, React, Node.js, client-server, Yahoo Finance API, data visualization, financial data, Express, user interface.

Довідки	Формат	Значення	Найменування	Кіл. листів	№ елементів	Додаток
			Документація загальна Знову розроблена			
<i>A4</i>		<i>ІАЛЦ.467200.002 ТЗ</i>	Веб-застосунок для економічного аналізу азійського регіону Технічне завдання	4		
<i>A4</i>		<i>ІАЛЦ.467200.003 ПЗ</i>	Веб-застосунок для економічного аналізу азійського регіону Пояснювальна записка	74		
<i>A4</i>		<i>ІАЛЦ.467200.004 Д1</i>	Веб-застосунок для економічного аналізу азійського регіону Алгоритм формування біржових даних	1		
<i>A4</i>		<i>ІАЛЦ.467200.005 Д2</i>	Веб-застосунок для економічного аналізу азійського регіону Діаграма класів	1		
<i>A4</i>		<i>ІАЛЦ.467200.006 Д3</i>	Веб-застосунок для економічного аналізу азійського регіону Структурна модель системи	1		
<i>A4</i>		<i>ІАЛЦ.467200.007 Д4</i>	Веб-застосунок для економічного аналізу азійського регіону Текст програмного коду	9		

					<b>ІАЛЦ.467200.001 ОП</b>			
Зм.	Аркуш	№ докум.	Підпис	Дата				
Розробив		Гонтар Д. С.			<i>Веб-застосунок для економічного аналізу азійського регіону</i>	Літ.	Аркуш	Аркушів
Перевірив		Кулаков Ю. О.					1	1
Н. Контр.		Нікольский С.С.			<b>Опис альбому</b>	<b>КПІ ім. Ігоря Сікорського, ФІОТ, ІО-15</b>		
Затв.								

**ТЕХНІЧНЕ ЗАВДАННЯ  
НА ДИПЛОМНИЙ ПРОЄКТ  
на тему: «Веб-застосунок для  
економічного аналізу азійського  
регіону»**

## ЗМІСТ

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ.....	2
2. ПІДСТАВИ ДЛЯ РОЗРОБКИ.....	2
3. МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ.....	2
4. ДЖЕРЕЛА РОЗРОБКИ.....	2
5. ТЕХНІЧНІ ВИМОГИ.....	3
5.1 Вимоги до розробленого продукту.....	3
5.2 Вимоги до програмного забезпечення.....	3
5.3 Вимоги до апаратної частини.....	3
6. ЕТАПИ РОЗРОБКИ.....	4

					<b>ІАЛЦ.467200.002 ТЗ</b>					
Зм.	Аркуш	№ докум.	Підпис	Дата	<i>Веб-застосунок для економічного аналізу азійського регіону</i>  <b>Технічне завдання</b>			Літ.	Аркуш	Аркушів
Розробив	Гонтар Д. С..							1	4	
Перевірив	Кулаков Ю. О..									
Н. Контр.	Нікольский С.С.							КПІ ім. Ігоря Сікорського, ФІОТ, ІО-15		
Затв.										

## **1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ**

Це технічне завдання поширюється на дипломний проєкт бакалавра за темою "Вебзастосунок для аналізу економічних показників азійського регіону". Застосунок може бути використаний студентами, аналітиками, дослідниками та всіма, хто цікавиться економічною ситуацією в країнах Азії для візуалізації даних, прийняття рішень або подальших досліджень.

## **2. ПІДСТАВИ ДЛЯ РОЗРОБКИ**

Підставою для розробки системи є завдання на виконання бакалаврського проєкту, по освітньо-професійної програми «Комп'ютерні системи та мережі» спеціальності 123 «Комп'ютерна інженерія», затверджене кафедрою Обчислювальної техніки НТУ України «Київський Політехнічний інститут імені Ігоря Сікорського».

## **3. МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ**

Метою проєкту є розробка сучасного вебзастосунку, що дозволяє користувачам аналізувати структуру азійського ринку, розподіл компаній за секторами й країнами, а також переглядати показники, які характеризують вплив окремих компаній на економіку регіону.

## **4. ДЖЕРЕЛА РОЗРОБКИ**

Основними джерелами для розробки слугували: науково-технічна література, наукові статті на задану тему, публікації та відкриті дані в мережі Інтернет та офіційні документації.

					ІАЛЦ.467200.002 ТЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		2

## 5. ТЕХНІЧНІ ВИМОГИ

### 5.1 Вимоги до розробленого продукту

Програмне забезпечення повинно забезпечувати:

- Інтерактивне відображення біржової інформації по країнах Азії;
- Наявність сторінок з детальними даними про кожну країну та сектор економіки;
- Відображення графіків та діаграм
- Реалізацію ефективної навігації між сторінками;
- Відображення аналітичних матеріалів
- Інтеграцію з реальними зовнішніми джерелами даних
- Відображення інтерактивного картографічного компоненту

### 5.2 Вимоги до програмного забезпечення

Необхідні компоненти для запуску продукту:

- Node.js (версія  $\geq 18.10$ ), Express;
- Інструменти розробки: Visual Studio Code, Git;
- Tailwind CSS, Recharts, Axios.;
- ОС: Windows / macOS / Linux.

### 5.3 Вимоги до апаратної частини

Мінімальні вимоги для роботи клієнтської частини:

- Оперативна пам'ять: від 4 ГБ;
- Підключення до мережі Інтернет.

					ІАЛЦ.467200.002 ТЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		3

## 6. ЕТАПИ РОЗРОБКИ

№	Назва етапу	Термін виконання
1	Аналіз потреб користувачів і формування вимог	
2	Проектування архітектури системи (frontend/backend)	
3	Реалізація серверної частини з використанням Django REST Framework і PostgreSQL	
4	Створення клієнтської частини за допомогою Next.js, TypeScript та Material UI	
5	Налаштування середовища розробки та запуску (Docker, docker-compose)	
6	Інтеграція з ApexCharts для візуалізації фінансових показників	
7	Тестування компонентів системи	
8	Створення API-документації за допомогою Swagger і ReDoc	
9	Розгортання системи, налаштування pre-commit хуків для забезпечення якості коду	
10	Підготовка супровідної технічної документації	

					ІАЛЦ.467200.002 ТЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		4

**ПОЯСНЮВАЛЬНА ЗАПИСКА  
НА ДИПЛОМНИЙ ПРОЄКТ  
на тему: «Веб-застосунок для  
економічного аналізу азійського  
регіону»**

Київ - 2025

ПЕРЕЛІК СКОРОЧЕНЬ .....	3
ВСТУП .....	4
РОЗДІЛ 1. Огляд існуючих рішень для економічного аналізу .....	6
1.1 Сучасний стан інформаційних систем економічного аналізу .....	6
1.2 Огляд популярних вебзастосунків для економ аналізу Азії та інших регіонів .....	7
1.3 Недоліки сучасних систем економічного аналізу для дослідження азійського регіону .....	8
1.4 Недоліки сучасних систем економічного аналізу для дослідження азійського регіону .....	8
ВИСНОВКИ ДО РОЗДІЛУ 1 .....	10
РОЗДІЛ 2. Огляд та вибір технологій для розробки системи .....	11
2.1 Аналіз сучасних підходів до створення вебзастосунків .....	11
2.2 Обґрунтування вибору технологічного стеку серверної частини .....	12
2.3 Node.js та Express .....	12
2.4 Технологічний стек клієнтської частини .....	14
2.4.1. Вибір React .....	15
2.4.2 Візуалізація даних: Chart.js та react-simple-maps .....	16
2.4.3 Інтеграція з фінансовим API: Yahoo Finance API та бібліотека yahoo-finance2 .....	18
2.5 Допоміжні інструменти: Axios, Nodemon, Tailwind CSS .....	21
ВИСНОВКИ ДО РОЗДІЛУ 2 .....	24
РОЗДІЛ 3. ДЕТАЛІ РОЗРОБКИ СИСТЕМИ .....	26
3.1 Реалізація серверної частини .....	26
3.1.1 Головний файл server.js .....	28
3.1.2 Основні контролери .....	30
3.1.3 Основні контролери .....	30
3.1.4 Конвертація валют .....	33
3.1.5 Фінансові дані .....	34
3.1.6 Кешування та структура файлу realCompanies.json .....	35
3.2 Реалізація клієнтської частини .....	37
3.2.1 Домашня сторінка .....	38
3.2.2 Навігація між сторінками .....	41
3.2.3 Сторінки окремих країн .....	43
3.2.4 Сторінка Research.jsx .....	46
ВИСНОВКИ ДО РОЗДІЛУ 3 .....	48

					<b>ІАЛЦ.467200.003 ПЗ</b>			
Зм.	Аркуш	№ докум.	Підпис	Дата	<i>Веб-застосунок для економічного аналізу азійського регіону</i>	Літ.	Аркуш	Аркушів
Розробив		Гонтар Д. С.					1	74
Перевірив		Кулаков Ю. О.						
Н. Контр.		Нікольский С.С				КПІ ім. Ігоря Сікорського, ФІОТ, ІО-15		
Затв.					<b>Технічне завдання</b>			

РОЗДІЛ 4. Дослідження та аналіз розробленої системи .....	49
4.1 Загальна оцінка працездатності системи .....	49
4.2 Перевірка функціональності ключових сторінок .....	50
4.2.1. Сторінка Home .....	50
4.2.2 Сторінка CountryPage .....	52
4.2.3 Сторінка SectionPage .....	54
4.2.4 Сторінки огляду Countries і Sections .....	56
4.2.5 Сторінка Research .....	58
4.2.6 Навігація .....	59
4.3 Оцінка швидкодії та стабільності .....	61
4.4 Аналіз відповідності технічним вимогам .....	63
ВИСНОВКИ ДО РОЗДІЛУ 4 .....	66
ВИСНОВКИ .....	68
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	71

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		2

## ПЕРЕЛІК СКОРОЧЕНЬ

API	(Application Programming Interface) Прикладний програмний інтерфейс
ACID	(Atomicity, Consistency, Isolation, Durability) – Атомарність, узгодженість, ізоляція, довговічність
CORS	(Cross-Origin Resource Sharing) – Міждоменний обмін ресурсами
CRUD	(Create, Read, Update, Delete) – Створення, читання, оновлення, видалення
CSS	(Cascading Style Sheets) – Каскадні таблиці стилів
DOM	(Document Object Model) – Об'єктна модель документа
HTTP	(Hypertext Transfer Protocol) – Протокол передачі гіпертексту
JIT	(Just-In-Time) – Компіляція "точно в строк"
JSON	(JavaScript Object Notation) – Об'єктна нотація JavaScript
JSX	(JavaScript XML) – Розширення синтаксису JavaScript
MPA	(Multi-Page Application) – Багатосторінковий додаток
MVP	(Minimum Viable Product) – Мінімально життєздатний продукт
npm	(Node Package Manager) – Менеджер пакетів Node.js
PaaS	(Platform as a Service) – Платформа як послуга
REST	(Representational State Transfer) – Передача репрезентативного стану
SEO	(Search Engine Optimization) – Пошукова оптимізація
UX	(User Experience) – Користувацький досвід

## ВСТУП

Актуальність створення нового вебзастосунку зумовлена нагальною потребою у зручному, доступному та спеціалізованому інструменті для глибокого вивчення економіки азійського регіону. Сучасний глобальний економічний ландшафт дедалі більше формується завдяки динамічному зростанню та еволюційній складності азійських ринків. Розуміння цих ринків вимагає інструментів, що виходять за рамки загальних платформ для даних, пропонуючи спеціалізовані можливості для тонкого аналізу. Запропонований вебзастосунок має на меті заповнити цю критичну прогалину, надаючи зручний, доступний та спеціалізований інструмент для дослідження азійської економіки.

Проектований застосунок є особливо корисним у навчальному процесі, оскільки він спрощує роботу з великими обсягами економічної інформації, дозволяє проводити порівняльний аналіз компаній та сприяє розвитку аналітичних навичок користувачів. Це підкреслює його роль як інструменту, що покращує освітні можливості та сприяє глибшому розумінню економічних концепцій, подібно до того, як це роблять інші цифрові навчальні платформи [1]. Наприклад, цифрові ресурси та онлайн-платформи трансформують викладання економіки, пропонуючи інтерактивні симуляції та форуми для дискусій, що сприяє активному навчанню та критичному мисленню [1]. Азійська економічна панель (Asian Economic Panel), наприклад, зосереджується на ретельному аналізі ключових економічних питань окремих азійських економік та ширшого азійського регіону, що підкреслює академічну спрямованість цієї галузі.

Швидка трансформація та зростаючий вплив азійських економік вимагають динамічної аналітичної платформи, яка може адаптуватися до мінливих економічних структур та складності даних. Це означає, що дизайн застосунку має бути гнучким та здатним інтегрувати нові типи

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		4

даних та аналітичні моделі в міру розвитку регіону. Така гнучкість є критично важливою, оскільки економічні умови та доступність даних в Азії постійно змінюються, і статичний інструмент швидко втратить свою актуальність. Отже, розробка власного рішення дозволяє зосередитися на специфіці азійських ринків, запропонувати інтуїтивно зрозумілий інтерфейс та забезпечити користувача інтерактивними інструментами для аналізу компаній за секторами, країнами та іншими критеріями.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		5

# РОЗДІЛ 1. Огляд існуючих рішень для економічного аналізу

## 1.1 Сучасний стан інформаційних систем економічного аналізу

На сучасному етапі розвитку інформаційні системи для економічного аналізу становлять складні технологічні комплекси, покликані забезпечувати ефективний збір, обробку, аналіз та візуалізацію значних обсягів економічних даних. В умовах невідомого зростання інформаційних потоків та різноманітності джерел даних, подібні системи стають незамінними інструментами для ухвалення обґрунтованих управлінських та стратегічних рішень в економічній сфері.

Сучасні платформи економічного аналізу здійснюють інтеграцію різноманітних категорій даних. До них належать фінансові показники діяльності компаній, динаміка ринків цінних паперів, відомості про корпоративні події, а також ключові макроекономічні параметри, такі як індекси споживчих цін, показники валового внутрішнього продукту, рівень зайнятості та інфляція [1]. Часто такі платформи оснащені потужними засобами візуалізації, що дають змогу створювати інтерактивні графіки, діаграми та географічні карти, що істотно спрощує сприйняття складних економічних процесів.

Важливим вектором розвитку сучасних інформаційних систем є застосування технологій Big Data та штучного інтелекту (AI). Ці технології дозволяють здійснювати глибокий аналіз значних масивів неструктурованих даних, отриманих з нетрадиційних джерел, таких як соціальні мережі, веб-ресурси та мобільні пристрої [2]. Це забезпечує значне підвищення точності прогнозування різноманітних економічних явищ, включаючи кризові ситуації, міграційні тенденції, демографічні

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		6

зміни та реакцію ринків на зовнішні фактори [2, 3].

## **1.2 Огляд популярних вебзастосунків для економ аналізу Азії та інших регіонів**

В даний час існує широкий спектр вебзастосунків та платформ, що надають користувачам інструменти для економічного аналізу та досліджень. Серед найбільш відомих міжнародних сервісів можна виділити наступні:

- International Financial Statistics (IFS) – платформа Міжнародного валютного фонду, яка надає комплексну інформацію про макроекономічні показники країн світу, включаючи валовий внутрішній продукт, інфляцію, валютні курси, міжнародну торгівлю та фінансову діяльність [4].
- World Development Indicators (WDI) – платформа Світового банку, яка пропонує широкий вибір даних щодо економічного розвитку, населення, довкілля, а також соціальних та інфраструктурних показників понад 200 країн світу [5].
- Bloomberg Terminal – професійна система для аналізу фінансових ринків, що використовується фінансовими установами, корпораціями та дослідниками для отримання актуальних даних про акції, облігації, валютні ринки, товарні біржі та економічні новини [6].
- Trading Economics – ресурс, який надає статистичні дані, графіки та аналітичні матеріали по економіці різних країн, у тому числі азійських. Сервіс дозволяє зручно порівнювати економічні показники різних держав, здійснювати прогнозування та аналізувати динаміку економічних процесів [7].

Попри розмаїття та широкі можливості зазначених платформ, вони мають певні обмеження, особливо в контексті глибокого аналізу специфіки

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ док.ум.	Підпис	Дата		7

економічного розвитку азійського регіону.

### **1.3 Недоліки сучасних систем економічного аналізу для дослідження азійського регіону**

КНезважаючи на значний прогрес у розвитку технологій аналізу даних, існуючі системи економічного аналізу все ще мають ряд обмежень, що впливають на повноту та якість дослідження економіки Азії. Однією з ключових проблем є недостатня деталізація та агрегація даних, що призводить до втрати важливих унікальних особливостей окремих країн та секторів. Це значною мірою ускладнює точне розуміння внутрішньорегіональних економічних процесів [8].

Крім того, для багатьох азійських країн характерною є проблема доступності та достовірності офіційних статистичних даних. Політичні чинники часто можуть спотворювати об'єктивність економічних показників, змушуючи дослідників звертатися до альтернативних джерел, таких як супутникові знімки, обсяги споживання електроенергії та логістичні дані, які можуть більш точно відображати реальний економічний стан [9].

Ще одним викликом є відсутність єдиних стандартів та уніфікованих підходів до обробки та візуалізації економічних даних. Це суттєво ускладнює порівняльний аналіз між країнами та секторами. Різноманітність форматів даних, методів збору та інтерпретації інформації потребує спеціалізованих рішень, які були б адаптовані до конкретних умов азійського регіону.

### **1.4 Недоліки сучасних систем економічного аналізу для дослідження азійського регіону**

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		8

Враховуючи виявлені недоліки, вебзастосунок, що розробляється, має забезпечити точний та детальний аналіз економічної ситуації в країнах Азії. Основними вимогами до системи є:

- Реалізація зручних інструментів для динамічної фільтрації даних за країнами, секторами та фінансовими показниками, що дозволяє детально вивчати специфіку кожного регіону.
- Забезпечення інтерактивної візуалізації даних, яка включає динамічні графіки, карти, таблиці та інші елементи, що сприяють швидкому розумінню економічних тенденцій [7, 9].
- Використання сучасних технологій (React [16], Node.js [12], API Yahoo Finance [21] та інших ресурсів) для збору та обробки актуальних даних у реальному часі, що гарантує їх достовірність та актуальність.

Таким чином, розроблений вебзастосунок вирішуватиме основні проблеми існуючих систем економічного аналізу та дозволить користувачам здійснювати точний, глибокий та інтерактивний аналіз економіки азійського регіону.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		9

# ВИСНОВКИ ДО РОЗДІЛУ 1

У першому розділі бакалаврської роботи було здійснено аналіз сучасного стану інформаційних систем, що застосовуються для економічного аналізу. Виявлено, що сучасні платформи демонструють широкий спектр функціональних можливостей щодо збору, обробки та візуалізації різноманітних економічних даних. Вони активно використовують як традиційні статистичні джерела [1, 4, 5, 6, 7], так і сучасні методи роботи з великими даними, включаючи інтеграцію інформації з альтернативних джерел для підвищення точності й глибини аналізу [2, 3, 9].

Огляд популярних вебзастосунків та цифрових платформ показав їхню ефективність у роботі з макроекономічними показниками та корпоративними даними. Проте, було визначено низку обмежень, які суттєво впливають на якість та глибину аналізу в контексті Азії. До таких обмежень належать надмірна агрегація даних, сумнівна достовірність офіційних джерел інформації, особливо в країнах із потужним політичним впливом на статистику [8], а також технологічні та регуляторні труднощі.

Запропонований вебзастосунок має на меті усунути ці недоліки через впровадження детального механізму фільтрації даних, динамічної візуалізації та інтерактивних інструментів аналізу, що забезпечить детальний та точний економічний аналіз азійського регіону. Такий підхід дозволить ефективніше досліджувати особливості та внутрішню динаміку азійських економік, сприяючи кращому розумінню та ухваленню обґрунтованих рішень користувачами системи. Водночас, визначені нефункціональні вимоги гарантуватимуть стабільність, безпеку та зручність використання вебзастосунку у довгостроковій перспективі..

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		10

## РОЗДІЛ 2.Огляд та вибір технологій для розробки системи

### 2.1 Аналіз сучасних підходів до створення вебзастосунків

Сучасні вебзастосунки будуються на принципах розподіленої багаторівневої архітектури, що забезпечує гнучкість та масштабованість у порівнянні зі старішими монолітними системами [10, 11]. Традиційна монолітна архітектура поєднує інтерфейс, бізнес-логіку та доступ до даних в одному застосунку, що ускладнює розвиток і масштабування. Натомість сучасні підходи передбачають чітке розділення на шари: клієнтський інтерфейс (frontend), серверну частину (backend) та базу даних. Це дозволяє змінювати або розгортати кожен компонент незалежно, не впливаючи на інші, що підвищує гнучкість системи.

На боці клієнта широкого поширення набули односторінкові вебзастосунки (SPA). У SPA-архітектурі початково сервер відправляє мінімальну HTML-сторінку і JavaScript-код, який динамічно дорендерує вміст у браузері без повного перезавантаження сторінок [10]. Це покращує інтерактивність та швидкодію інтерфейсу, адже обмін даними з сервером відбувається асинхронно через API. Для порівняння, традиційні підходи з серверним рендерингом (SSR) генерують повну HTML-сторінку на кожний запит, що може сповільнювати взаємодію при високому навантаженні. Тенденція до SPA добре узгоджується з концепцією headless-архітектури, коли фронтенд працює окремо від бекенду через чітко визначені API [11]. Такий поділ полегшує підтримку і дає змогу розробляти клієнтську частину незалежно (або навіть кілька різних клієнтів, як-от веб та мобільний) без зміни логіки сервера.

Таким чином, сучасні підходи до веброзробки характеризуються розділенням системи на незалежні компоненти, асинхронною взаємодією

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		11

між ними та використанням SPA на фронтенді. Це забезпечує застосунок високу масштабованість, гнучкість і продуктивність у порівнянні зі старішими монолітними рішеннями. Обраний стек технологій у дипломному проєкті відповідає цим тенденціям: окремий SPA-фронтенд, полегшений Node-бекенд і чіткий контракт REST.

## **2.2 Обґрунтування вибору технологічного стеку серверної частини**

Враховуючи аналіз підходів, для реалізації бекенду проєкту було вирішено використати платформу Node.js з фреймворком Express. На вибір вплинули особливості архітектури Node.js, його продуктивність при обробці великих обсягів запитів та можливість використовувати одну мову програмування (JavaScript) і на сервері, і в клієнті.

## **2.3 Node.js та Express**

Node.js – це середовище виконання JavaScript на боці сервера, побудоване на базі високопродуктивного двигуна V8 від Google. Головною архітектурною відмінністю Node.js є однопотокова, асинхронна модель з циклом подій [12]. Замість запуску нового потоку для кожного клієнтського запиту (як це роблять типові сервери на Java чи .NET), Node.js обробляє усі запити в межах одного потоку, використовуючи неблокуючий ввід/вивід та чергу подій. Це означає, що поки одна операція очікує операції вводу-виводу (наприклад, відповіді від API чи читання з бази даних), потік може переключитися на обробку іншого запиту. Такий підхід дозволяє Node.js підтримувати тисячі одночасних з'єднань, економно витрачаючи пам'ять та ресурси. За умов, коли запити не надто важкі обчислювально, продуктивність Node.js проявляється найкраще –

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		12

платформа забезпечує високу пропускну здатність і низькі затримки при інтенсивному мережевому трафіку. Автори Node.js першочергово проектували його під задачі реального часу, де потрібні постійні двонаправлені з'єднання (наприклад, чати, стрімінг даних) [12]. Для нашого застосунку (отримання фінансових даних у режимі, наближеному до реального часу, та обслуговування запитів від клієнтів) така модель є надзвичайно доречною.

Використання Node.js також сприяє уніфікації технологій на проєкті. Оскільки фронтенд написано на JavaScript (React), вибір Node.js для бекенду означає, що вся розробницька команда працює з однією мовою [12]. Це підвищує ефективність: розробникам легше переключатися між клієнтською і серверною частинами, перевикористовувати код та узгоджувати інтерфейси. Дослідження відзначають, що спільна мова фронт- і бекенду дозволяє формувати більш крос-функціональні команди та прискорює доставку продукту. У випадку навчального проєкту, де обмежені ресурси, це особливо важливо: знання JavaScript є достатнім для реалізації повного стеку, що знижує поріг входження для нових учасників та потенційно зменшує час розробки.

Express – це мінімалістичний веб-фреймворк для Node.js, який забезпечує базовий каркас для створення веб-серверу [13]. Його філософія – бути гнучким, швидким та не обмежувати розробників у виборі архітектурних рішень для веб-застосунків на Node.js. Express не нав'язує жорсткої структури проєкту (на відміну від деяких великих фреймворків, як-от Django чи Angular на фронтенді) – натомість він надає тільки необхідний мінімум: гнучку систему маршрутизації URL, підтримку шаблонів (опціонально) та механізм middleware для обробки запитів і відповідей [13]. Така ненав'язливість (unopinionated framework) дозволяє будувати архітектуру бекенду так, як вимагає проєкт, додаючи лише потрібні компоненти. Express має багату екосистему middleware-пакетів,

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		13

доступних через npm, які реалізують автентифікацію, логування, обробку помилок, підтримку сесій тощо. В результаті можна зібрати саме ту конфігурацію, що потрібна для конкретного випадку, не обтяжуючи систему зайвими модулями. У нашому проєкті Express використовується для реалізації RESTful API: визначено набори маршрутів для отримання економічних даних, використано middleware для обробки JSON-запитів та налаштування заголовків безпеки.

Вибір Node.js + Express був також зіставлений з альтернативами. Розглядалися, зокрема, Python з Flask/Django та інші серверні платформи. Flask (Python) є аналогічним мінімалістичним фреймворком і добре підходить для швидкого прототипування, а Django пропонує “батареї в комплекті” (ORM, адмін-інтерфейс, автентифікація) для швидкого створення складних застосунків [14]. Однак, Python-фреймворки працюють у багатопотоковому режимі з блокуючим виконанням за замовчуванням. Це означає, що обробка великої кількості одночасних запитів може вимагати більше ресурсів або асинхронних додаткових засобів (наприклад, використання ASGI/FastAPI). Node.js в таких сценаріях демонструє вищу ефективність на запити вводу-виводу завдяки неблокуючій, подієво-орієнтованій моделі [12]. Дослідження продуктивності показують, що при великій кількості паралельних з’єднань Node.js перевершує Django за швидкістю відповіді та використанням ресурсів [14]. Для задач реального часу (стрімінгові дані, часті оновлення), до яких належить і наш застосунок, Node.js є кращим вибором. Крім того, залишаючись в екосистемі JavaScript, легше інтегрувати сторонні бібліотеки (зокрема для фінансових API) та підтримувати єдиний стиль кодової бази.

## 2.4 Технологічний стек клієнтської частини

Клієнтська частина (frontend) є надзвичайно важливою для

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		14

вебзастосунку, особливо коли мова йде про відображення економічних даних та надання інструментів для їх аналізу. В цьому проекті фронтенд реалізовано у вигляді односторінкового додатка (SPA) з використанням бібліотеки React. Такий вибір обумовлений вимогами до інтерфейсу: динамічне оновлення графіків, інтерактивні карти регіону, швидкий відгук на дії користувача. React на сьогодні є одним з найпопулярніших інструментів для побудови UI, і його особливості добре підходять під зазначені вимоги [17, 18]. Крім React, для реалізації специфічних потреб було обрано бібліотеки Chart.js (для графіків) та react-simple-maps (для географічних карт). Нижче наведено аналіз переваг React над альтернативами та обґрунтування використання згаданих бібліотек для візуалізації даних..

### 2.4.1. Вибір React

React – це JavaScript-бібліотека для розробки інтерфейсу користувача, створена компанією Facebook (Meta). В її основі – компонентний підхід: інтерфейс розбивається на дрібні повторно використовувані компоненти, кожен з яких інкапсулює власну логіку відображення і стан [16, 17]. Це значно спрощує розробку та підтримку складних інтерфейсів: зміни в одному компоненті мінімально впливають на інші, код легше тестувати і масштабувати. React запровадив концепцію віртуального DOM – ефективного в пам’яті представлення структури сторінки, яке дозволяє оптимізувати операції оновлення реального DOM [18]. При зміні стану компонентів React обчислює “де саме” потрібно внести зміни у справжній DOM і робить це точково, а не повністю перерендерює всю сторінку. Це забезпечує високу продуктивність інтерфейсу навіть при частих оновленнях даних, що актуально, наприклад, для оновлення фінансових графіків у режимі реального часу.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		15

Однією з головних причин популярності React є його гнучкість та багатство екосистеми [16]. React часто називають “бібліотекою, а не фреймворком”, оскільки він зосереджується тільки на шарі представлення (V у парадигмі MVC). Розробник має свободу вибору, які інструменти додати для управління станом (Redux, MobX чи Context API), для маршрутизації (React Router) тощо [16, 17]. Ця невимушеність спрощує інтеграцію React у будь-який проєкт: можна поступово впроваджувати React в існуючий код або обрати лише потрібні додаткові бібліотеки. Для нашого застосунку це означає, що ми змогли легко підключити специфічні пакети для графіків і карт без конфліктів з основною бібліотекою.

У контексті вебзастосунку для економічного аналізу React надає кілька конкретних переваг. По-перше, компонентний підхід дозволяє ізолювати логіку окремих візуалізацій: є компонент графіку, компонент мапи, компонент таблиці показників тощо. Вони можуть повторно використовуватися для різних сторінок або секцій застосунку, що зменшує обсяг коду і помилок. По-друге, активна екосистема React означає наявність готових бібліотек для майже будь-яких потреб – зокрема, ми скористалися готовими рішеннями для роботи з фінансовими API та побудови графіків, що інтегруються з React безперешкодно. По-третє, React чудово підходить для побудови інтерактивних інтерфейсів: через механізм стану (state) ми легко реалізуємо фільтри, перемикачі показників, масштабування діаграм – усе це оновлює представлення миттєво, без перезавантаження сторінки, підвищуючи зручність роботи аналітика

#### 2.4.2 Візуалізація даних: Chart.js та react-simple-maps

Для успішного економічного аналізу недостатньо отримати числові дані – їх треба ще наочно представити у вигляді графіків, діаграм, карт. У нашому застосунку вирішено використати перевірені часом бібліотеки для

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ док.ум.	Підпис	Дата		16

візуалізації даних: Chart.js для побудови діаграм (часові ряди, гістограми, секторні діаграми тощо) та React-simple-maps для відображення картографічної інформації (мапа азійського регіону з виділенням показників по країнах). Обидва інструменти добре інтегруються з React і забезпечують високу якість графічного представлення при помірних затратах на розробку. Розглянемо їх детальніше та обґрунтуємо їх вибір.

Chart.js – це популярна відкрита бібліотека JavaScript для створення інтерактивних діаграм, що існує з 2013 року і підтримується спільнотою розробників [19]. Вона підтримує щонайменше вісім основних типів графіків “із коробки”: лінійні, стовпчикові, площинні (area), кругові (секторні), бульбашкові, радарні, полярні та розсіювання (scatter) [19]. Такий набір покриває практично всі типові потреби при візуалізації економічних даних – від динаміки ВВП (лінійний графік) до структури експорту по країнах (секторна діаграма). Chart.js побудований на основі елементу HTML5 Canvas, що дозволяє рендерити графіку на стороні клієнта дуже ефективно і з високою якістю (графіки виглядають чітко на різних роздільностях екрана). Значною перевагою є простота використання: щоб створити графік, достатньо передати бібліотеці дані та параметри, все малювання та взаємодію (наприклад, спливаючі підказки при наведенні) Chart.js бере на себе. Бібліотека надає гнучкі можливості налаштування стилів, кольорів, підписів осей, а також підтримує плагіни – наприклад, плагін анімації чи збереження графіка у файл.

React-simple-maps – це спеціалізована бібліотека для побудови карт на базі React. Вона створена для того, щоб зробити роботу з SVG-картами простою і декларативною у середовищі React [20]. В основі react-simple-maps лежать напрацювання D3 (зокрема, модулі d3-geo для географічних проєкцій та TopoJSON для опису карт), але все складне налаштування вже вирішено всередині бібліотеки. Розробник отримує набір готових React-компонентів – таких, як <ComposableMap>, <Geographies> і <Geography> –

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		17

які можна скомпонувати для відображення карти [20]. Бібліотека бере на себе рутинні задачі: завантаження і парсинг даних карти (ТороJSON файлів), проектування координат на площину, масштабування, панорамування (пересування карти) та навіть оптимізацію рендерингу при взаємодії. Фактично, react-simple-maps дозволяє отримати інтерактивну SVG-карту з можливістю масштабувати та перетягувати її з мінімальним кодом. Важливо, що ця бібліотека побудована з урахуванням React-підходу: вона передає управління DOM самому React, тому легко поєднується з іншими React-бібліотеками.

Отже, поєднання Chart.js та react-simple-maps покриває потреби нашого фронтенду у візуалізації. Chart.js надає простий спосіб генерувати традиційні діаграми, що є незамінним при відстеженні динаміки економічних індикаторів. React-simple-maps, у свою чергу, розв'язує задачу відображення даних у географічному розрізі, роблячи інтерфейс нашого застосунку більш наочним та інтуїтивним. Обидва рішення добре зарекомендували себе у веброботці: Chart.js широко використовується і розглядається як одна з найкращих бібліотек для візуалізації даних у JS, а react-simple-maps хоч і більш спеціалізована, але спирається на перевірені технології D3 та SVG, що гарантує якісний результат. Вибір цих інструментів дозволив нам не “винайти велосипед”, а використати переваги готових бібліотек, сконцентрувавшись на предметній логіці аналізу даних.

### **2.4.3 Інтеграція з фінансовим API: Yahoo Finance API та бібліотека yahoo-finance2**

Однією з ключових функціональних вимог нашого застосунку є можливість отримувати актуальні фінансові та економічні дані для аналізу. Зокрема, йдеться про котирування акцій азійських компаній, значення

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		18

фондових індексів, валютні курси, а також фінансові показники (такі як прибутки компаній, капіталізація тощо). На сьогодні існує багато платних і безкоштовних джерел таких даних – так званих фінансових API. Для реалізації дипломного проекту було вирішено скористатися даними Yahoo Finance – відомого веб-порталу, що містить фінансову інформацію по світових ринках. Компанія Yahoo офіційно не надає відкритого API для сторонніх розробників, проте ентузіасти з спільноти підтримують неофіційні рішення для доступу до цих даних [21, 22]. Одним із таких рішень є пакет yahoo-finance2 для Node.js, який і було інтегровано в наш бекенд.

Yahoo Finance API (неофіційний) відзначається тим, що охоплює великий спектр фінансових інструментів і показників: акції, індекси, біржові фонди (ETF), валютні курси, ф'ючерси, опціонні дані, новини компаній та багато іншого [30]. Для нашої системи особливо цінним є те, що Yahoo Finance містить дані по багатьох азійських ринках, включно з фондовими біржами Японії, Китаю, Південної Кореї, Індії тощо. Використання Yahoo Finance дозволяє нам отримувати історичні ряди цін акцій, поточні біржові котирування в режимі реального часу, інформацію про капіталізацію, обсяг торгів, фінансову звітність компаній і навіть економічні індикатори (наприклад, курси валют чи ціни на сировину) – усе з одного джерела [30]. Важливо підкреслити, що Yahoo Finance є безкоштовним у використанні: запити до нього не потребують API-ключів чи оплати, хоча й не мають гарантій SLA (можливі обмеження на частоту запитів та зміни у форматі даних, оскільки API неофіційний).

Бібліотека yahoo-finance2 – це пакунок для Node.js, що фактично є неофіційним API-клієнтом Yahoo Finance, підтримуваним спільнотою розробників (автори Gadisc та Pilwon Huh) [21]. Ця бібліотека прийшла на зміну старішому пакету yahoo-finance (v1) і пропонує оновлений інтерфейс. Її ціль – надати прості у використанні функції для отримання різних видів

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		19

фінансової інформації з сайту [finance.yahoo.com](http://finance.yahoo.com). Yahoo-finance2 підтримує Node.js останніх версій і написана з використанням TypeScript, що гарантує типобезпеку при використанні [21]. Важливо зазначити, що розробники прямо вказують: проект не створено і не підтримується Yahoo Inc., і використання бібліотеки відбувається на власний ризик. Водночас на практиці ця бібліотека успішно працює вже багато років (попередник існував ще з 2013 р.), а спільнота швидко адаптується до змін у внутрішніх API Yahoo, тож для цілей дипломного проекту такий підхід цілком виправданий.

Функціональні можливості yahoo-finance2 дуже широкі. Вона надає методи для виконання запитів різних типів, зокрема: quote (поточні дані по інструменту – ціна, денний діапазон, попередній закриття тощо), historical (історичні котирування за датами), chart (дані для побудови графіку котирувань, схоже до historical, але з гнучкішим інтерфейсом), search (пошук тикера за назвою компанії), quoteSummary (детальна інформація з фінансових звітів, ключових показників і новин) та багато інших [21]. Фактично через quoteSummary можна отримати фінансову статистику компанії: баланси, кеш-фло, тренди прибутків, коефіцієнти P/E, боргові зобов'язання тощо – все, що відображається на сторінці Yahoo Finance для цієї компанії. Це відкриває широкі можливості для економічного аналізу. У нашому проекті, щоправда, ми поки що обмежилися основними видами даних: поточними котируваннями та історичними рядами цін, необхідними для побудови графіків динаміки. Та закладено потенціал і для розширення – наприклад, додати модуль аналізу фундаментальних показників компаній, оскільки yahoo-finance2 дозволяє це зробити запитом потрібних модулів (assetProfile, financialData, earnings тощо) [21].

Отже, інтеграція з Yahoo Finance через yahoo-finance2 є раціональним вибором, який підтвердив свою ефективність. Ми отримали єдиний, уніфікований канал даних для різних фінансових об'єктів,

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		20

безкоштовний та достатньо повний по покриттю [22]. Бібліотека `yahoo-finance2` значно спростила процес – фактично, зведений до виклику готових методів з боку `Node.js`-коду [22]. Такий підхід дозволяє зосередитися на аналізі отриманих даних, не витрачаючи зайвого часу на низькорівневі деталі HTTP-запитів або парсингу HTML. В результаті, наш застосунок може в реальному часі завантажувати і оновлювати необхідну інформацію про азійські ринки, що робить його функціональним та актуальним інструментом для економічного аналізу регіону.

## 2.5 Допоміжні інструменти: `Axios`, `Nodemon`, `Tailwind CSS`

Окрім основних технологій фронтенду та бекенду, у проекті використано низку допоміжних інструментів, які покращують процес розробки та якість фінального продукту. Кожен з цих інструментів відіграє свою роль у стеку технологій, сприяючи зручності написання коду, його тестуванню, стилізації інтерфейсу тощо. Нижче перелічено ключові з них та обґрунтовано їх застосування:

`Axios` – це популярна бібліотека для виконання HTTP-запитів у `JavaScript`, яку ми залучили як на клієнтській частині, так і на серверній [23]. Вона є `promise`-базованим HTTP-клієнтом для браузера і `Node.js`, що на практиці означає єдиний інтерфейс для здійснення AJAX-запитів незалежно від оточення. В браузері `Axios` внутрішньо використовує `XMLHttpRequest` або `Fetch API` для відправки запитів, а в `Node.js` – вбудований модуль `http`. Основні переваги `Axios`: зручний синтаксис (методи `.get()`, `.post()` тощо повертають об'єкт `Promise`), автоматичне перетворення відповідей у `JSON`, якщо сервер відправляє `JSON` (цю рутину `Axios` бере на себе), а також розширені можливості – налаштування таймаутів, автоматична обробка помилок, відміна запитів, перехоплювачі (`interceptors`) для додавання заголовків або логування [24]. У нашому

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		21

проекті Axios використовується на фронтенді для звернення до бекенд-API. Коли React-компоненту потрібно отримати дані (наприклад, завантажити свіжі фінансові показники), він викликає функцію з Axios і опрацьовує Promise. Axios значно спрощує цей код, роблячи його більш читабельним та надійним: у разі помилки досить обробити `.catch`, а не перевіряти статус вручну. Альтернативою могло б бути використання вбудованого `fetch()`, але Axios має переваги сумісності (підтримує старі браузері, де `fetch` може бути недоступний) та вбудоване краще керування помилками (відкидає Promise при статусі відповіді `>= 400`, що зручно) [24]. Також Axios дозволяє легко налаштувати базовий URL та заголовки для всіх запитів, що ми використали для встановлення базової адреси нашого API. У Node.js частині Axios може стати в нагоді для запитів до сторонніх сервісів – хоча ми переважно використовували `yahoo-finance2` для фінансових API, Axios лишається універсальним інструментом на випадок, якщо потрібно звернутися до інших REST-сервісів. Загалом, використання Axios прискорило розробку мережевої логіки і зробило код більш зрозумілим, що відповідає практикам сучасної веброботи.

Ще один інструмент, корисний у розробці сервера – це Nodemon. Він вирішує проблему автоматичного перезапуску Node.js-сервера при внесенні змін у вихідний код [26]. Nodemon моніторить файли проекту і якщо виявляє зміни, то перезапускає серверний процес Node.js. Це дуже зручно: розробнику не треба вручну зупиняти і запускати `node index.js` після кожної правки – достатньо зберегти файл, і Nodemon сама “перезбере” сервер і запустить його знову. У нашому проекті Nodemon інтегровано в процес розробки бекенду: команда `npm run dev-server` використовує `nodemon` для запуску файлу серверу. Таким чином, під час активної розробки будь-яка зміна (наприклад, додавання нового маршруту API або виправлення логіки) негайно набуває чинності – сервер перезавантажується і готовий приймати запити з оновленим кодом за

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		22

лічені секунди. Альтернативи: в останніх версіях Node.js з'явився вбудований режим `--watch`, що дає схожий функціонал без зовнішніх інструментів. Проте для сумісності і гнучкості ми використали Nodemon, оскільки він дозволяє гнучко налаштувати, які розширення файлів відслідковувати, які шляхи ігнорувати, а також може виконувати довільні команди перед рестартом. Ця гнучкість поки що вища, ніж у флагу `--watch`. Nodemon економить багато часу і зусиль розробників, що особливо відчутно у академічному проекті з обмеженим часом – можна зосередитися на коді, а не на рутинних діях. В продакшн-режимі Nodemon, звісно, не використовується (він встановлений як dev-залежність), тому на кінцевий додаток не впливає. Але на етапі розробки він став незамінним помічником, забезпечуючи швидкий цикл “зміна-код -> перезапуск -> тест” і підвищуючи продуктивність команди.

Окремо слід згадати про підхід до стилізації інтерфейсу. Замість використання класичних CSS-фреймворків (на кшталт Bootstrap) чи написання CSS з нуля, у проекті застосовано Tailwind CSS – сучасний “утилітарний” фреймворк стилів [27]. Tailwind дотримується принципу utility-first: він надає сотні дрібних CSS-класів, кожен з яких відповідає за окрему властивість стилю (наприклад, `text-center` для вирівнювання тексту по центру, `bg-blue-500` для синього фону певного відтінку, `mt-4` для верхнього відступу тощо) [28]. Ідея полягає в тому, що компонуючи ці класи прямо в HTML (JSX), розробник швидко створює дизайн будь-якої складності без написання кастомного CSS [29]. Tailwind CSS є низькорівневим і високо настроюваним фреймворком: за потреби, можна змінити тематику (кольорову палітру, розміри тощо) через конфігураційний файл, щоб він відповідав бренду чи задуму проекту. Але, на відміну від Bootstrap, Tailwind не містить готових компонентів з власними стилями – він не нав'язує жодних дизайнерських рішень, а тільки дає “будівельні блоки” для створення унікального дизайну [28].

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		23

## ВИСНОВКИ ДО РОЗДІЛУ 2

У другому розділі дипломної роботи було здійснено комплексний аналіз сучасних підходів до створення вебзастосунків та обґрунтовано вибір конкретного технологічного стеку для реалізації інтерактивного ресурсу економічного аналізу країн Азії. Основна архітектурна модель, покладена в основу розробки, — односторінковий застосунок (SPA), що, на відміну від класичних багатосторінкових рішень, забезпечує більш плавну взаємодію з інтерфейсом, миттєву реакцію на дії користувача та високу продуктивність за рахунок асинхронної обробки запитів. Такий підхід є доцільним для аналітичних систем, орієнтованих на роботу з динамічними даними.

Серверна частина реалізована на основі JavaScript-середовища Node.js, яке демонструє високу продуктивність у роботі з великою кількістю одночасних запитів завдяки подієво-орієнтованій, неблокуючій моделі виконання [12]. Використання Express.js як основного фреймворку дало змогу організувати структуру RESTful API, оптимізувати маршрутизацію, обробку запитів та логування [13]. Така архітектура забезпечує розширюваність, простоту інтеграції з іншими сервісами та підвищену підтримуваність у майбутньому.

Щодо зберігання даних, на етапі прототипування застосовувались локальні JSON-файли з попередньо зібраною інформацією, що дозволило оперативно перевірити основні функціональні сценарії. У фінальній версії застосунку реалізовано інтеграцію з зовнішніми API — зокрема, Yahoo Finance — для отримання актуальних ринкових показників у реальному часі, що значно підвищує аналітичну цінність ресурсу [21, 22, 30]. Обробка таких даних реалізується асинхронно та зберігається у внутрішній структурі застосунку в зручному JSON-форматі. На перспективу система має потенціал до переходу на повноцінну реляційну базу даних, таку як

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		24

PostgreSQL, що надає більше можливостей для складної аналітики, агрегації та забезпечення цілісності фінансових даних [15].

Клієнтська частина побудована з використанням бібліотеки React, яка забезпечує ефективне створення динамічного інтерфейсу, адаптивного до різних пристроїв [16, 17]. Її компонентна модель дозволяє гнучко масштабувати систему, повторно використовувати UI-елементи та легко підтримувати проєкт у довгостроковій перспективі.

Для реалізації інтуїтивно зрозумілої та ефективної візуалізації економічних даних використано бібліотеки Chart.js та react-simple-maps [19, 20]. Вони дозволяють створювати інтерактивні графіки, діаграми та карти, які є ключовими інструментами в інтерфейсі користувача для аналізу фінансової динаміки, розподілу компаній за країнами, секторами тощо.

В обміні між клієнтом і сервером застосовується бібліотека Axios — гнучкий засіб для HTTP-комунікації, що забезпечує обробку запитів, помилок, перехоплення і логіку автентифікації [23, 24]. Tailwind CSS — обраний інструмент для стилізації інтерфейсу — дозволив створити адаптивний, сучасний і візуально узгоджений дизайн у межах уніфікованого стилю, що поліпшує сприйняття даних і зменшує навантаження на користувача [27, 28, 29].

Таким чином, обраний стек технологій — JavaScript (Node.js + React), REST API, Yahoo Finance API, Chart.js / react-simple-maps / Tailwind CSS — дозволив реалізувати ефективний вебзастосунок, орієнтованого на динамічний економічний аналіз. Поєднання гнучкості, масштабованості, зручності розробки та відкритості використаних рішень створює фундамент для подальшого розвитку проєкту, його інтеграції з новими джерелами даних, розширення функціональності та підвищення аналітичної глибини у межах дослідження економіки Азії.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		25

## РОЗДІЛ 3. ДЕТАЛІ РОЗРОБКИ СИСТЕМИ

### 3.1 Реалізація серверної частини

Серверна частина реалізована на Node.js з використанням фреймворку Express [12, 13]. Вона організована у вигляді окремої директорії backend, яка містить основні файли сервера, маршрути, контролери, утиліти та дані. Замість традиційного використання баз даних, інформація про компанії зберігається у JSON-файлах; цей підхід спрощує етапи розробки та забезпечує високу портативність даних. Нижче наведено структуру файлів бекенду і призначення основних компонентів:

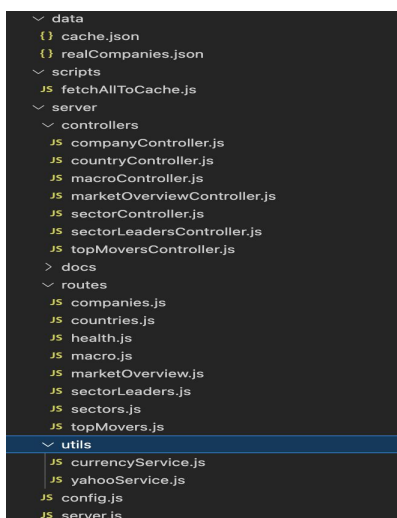


Рис 3.1 – Структура серверної частини

- server.js – це головний файл сервера, який ініціює роботу вебзастосунку. Він завантажує необхідні модулі, такі як Express та CORS, налаштовує маршрути й проміжне програмне забезпечення (middleware), а також запускає сервер на визначеному порті [13].
- routes/ – цей каталог містить файли з визначенням маршрутів Express. Кожен файл групує пов'язані endpoint-и API. У поточному проєкті,

наприклад, `companyRoutes.js` відповідає за маршрути, пов'язані з отриманням даних компаній. Маршрути підключаються у `server.js` з відповідними префіксами URL (наприклад, усі маршрути `companyRoutes` доступні за шляхом `/api/companies`).

- `controllers/` – у цьому каталозі знаходяться файли контролерів, що інкапсулюють основну логіку обробки вхідних запитів. Наприклад, `companyController.js` реалізує функції для отримання списку компаній, їхньої історії акцій та загальної статистики. Контролери відповідають за читання даних з локальних файлів, виконання необхідних обчислень або фільтрації та формування JSON-відповіді для клієнта.
- `data/` – цей каталог призначений для зберігання файлів даних у форматі JSON. Основним файлом є `realcompanies.json`, який містить попередньо зібрану інформацію про компанії (назва, країна, сектор, поточна ціна акції тощо). Контролери завантажують цей файл для доступу до базових даних. Додатково можуть використовуватися допоміжні файли, такі як `asia10.json` та `cache.json`, для кешування результатів запитів.
- `utils/` – каталог для утилітних модулів, що містять допоміжний код, який не прив'язаний безпосередньо до маршрутів Express. У даному проєкті тут може бути розташований модуль `currencyService.js`, який відповідає за роботу з валютами, наприклад, конвертацію цін акцій у спільну валюту та кешування валютних курсів. Цей модуль використовується контролерами для перетворення валют та може звертатися до зовнішнього API (якщо необхідно) для отримання актуальних курсів.
- `scripts/` – каталог, де розміщені скрипти для обробки даних. Прикладом є скрипт `fetchAllToCache.js`, призначений для отримання даних про компанії з зовнішніх джерел (таких як Yahoo Finance [21, 22]) та їх збереження у файл `Realcompanies.json`. Подібні скрипти виконуються окремо від основного сервера і слугують для початкового наповнення

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ док.ум.	Підпис	Дата		27

або періодичного оновлення даних.

### 3.1.1 Головний файл server.js

Файл server.js слугує ключовою точкою входу для серверної частини застосунку. У ньому відбувається ініціалізація Express-додатка, конфігурація глобального проміжного програмного забезпечення (middleware), підключення всіх визначених маршрутів та, зрештою, запуск HTTP-сервера. Розглянемо його структуру більш детально. На початку файлу server.js здійснюється імпорт необхідних модулів. На рисунку 3.2 представлено початковий етап ініціалізації серверного середовища у файлі server.js, що виступає основною точкою входу для бекенд-частини вебзастосунку

```
1  const express = require("express");
2  const cors = require("cors");
3  const path = require("path");
4  const fs = require("fs");
5
6  const cachePath = path.join(__dirname, "../data/cache.json");
7
8  // ROUTES
9  const companiesRouter = require("./routes/companies");
10 const sectorsRouter = require("./routes/sectors");
11 const countriesRouter = require("./routes/countries");
12 const macroRouter = require("./routes/macro");
13 const healthRouter = require("./routes/health");
14
15 // Додаткові аналітичні ендпоінти
16 const topMoversRouter = require("./routes/topMovers");
17 const sectorLeadersRouter = require("./routes/sectorLeaders");
18 const marketOverviewRouter = require("./routes/marketOverview");
19
20 const app = express();
21 const PORT = require("./config").PORT || 4000;
22
23 app.use(cors());
24 app.use(express.json());
```

Рис 3.2 – Імпорт модулів та налаштування базового середовища

Насамперед підключаються критично важливі модулі, які забезпечують базову функціональність: express для створення HTTP-сервера [13], cors для налаштування міждоменого обміну даними (що є особливо актуальним під час розробки та взаємодії з клієнтською

частиною), а також path та fs для ефективної роботи з файловою системою та формування абсолютних шляхів до таких ресурсів, як кешовані JSON-файли [12].

Наступним кроком є імпорт модулів маршрутів. Кожен з цих модулів відповідає за обробку певної логіки API, розділяючи функціональність на блоки: маршрути для компаній, країн, секторів економіки, макроекономічних показників, динаміки акцій, а також загальної ринкової аналітики. Така модульна організація архітектури значно підвищує підтримуваність проєкту та дозволяє чітко розмежувати відповідальність за різні типи даних між окремими компонентами [11].

На завершення цього фрагмента відбувається створення екземпляра Express-додатка, після чого підключаються два глобальні middleware. Ці налаштування створюють необхідну основу для безперебійної взаємодії між клієнтською частиною та сервером, забезпечуючи уніфіковану та ефективну структуру обробки запитів, що відповідає сучасним стандартам RESTful API [10]. На рисунку 3.3 візуалізовано процес підключення основних REST-маршрутів до серверного застосунку.

```
24 app.use("/api/companies", companiesRouter);
25 app.use("/api/sectors", sectorsRouter);
26 app.use("/api/countries", countriesRouter);
27 app.use("/api/macros", macroRouter);
28 app.use("/api/top-movers", topMoversRouter);
29 app.use("/api/sector-leaders", sectorLeadersRouter);
30 app.use("/api/market-overview", marketOverviewRouter);
31
32
33 app.get("/", (req, res) => {
34   res.send("Asian Economy Analysis API is running.");
35 });
36 app.use("/api/health", healthRouter);
37
38 app.use((err, req, res, next) => {
39   console.error("Server error:", err);
40   res.status(500).json({ error: "Server error" });
41 });
42
43 app.listen(PORT, () => {
44   console.log(`✅ Сервер запущено на порту ${PORT}`);
45 });
```

Рис 3.3 – Підключення маршрутів

Кожен виклик `app.use(...)` асоціює конкретний шлях (endpoint) з відповідним об'єктом маршрутизатора (Router), наприклад, `/api/companies` пов'язується з `companiesRouter`.

Цей підхід забезпечує ефективне розділення функціональності на логічні блоки, такі як компанії, сектори, країни, макроекономічні показники та лідери ринку [11]. Окрім основних API-маршрутів, конфігурація включає базовий GET / маршрут для тестування працездатності API та спеціальний обробник помилок (error-handling middleware). Запуск сервера виконується за допомогою `app.listen(...)` [13].

### 3.1.2 Основні контролери

Усі контролери сервера розташовані в каталозі `/server/controllers/` і виконують функцію логічної обробки даних для API. Кожен контролер обслуговує окремий маршрут та реалізує відповідну бізнес-логіку: від читання локальних файлів до взаємодії із зовнішніми сервісами, такими як Yahoo Finance API [21] та World Bank API [5]. Розглянемо два ключові контролери проєкту.

### 3.1.3 Основні контролери

Контролер `companyController.js` обслуговує маршрут `/api/companies` і відповідає за надання фінансової інформації про компанії. Його функціональність включає динамічне збагачення даних про компанії актуальними показниками, отриманими з Yahoo Finance API [21]. Рис 3.4. Фрагмент коду `companyController.js` Основна логіка контролера включає:

- Зчитування файлу `realCompanies.json`: З цього файлу завантажуються базові дані про компанії [23].

- Запити до `fetchCompanyData(ticker)`: Для кожної компанії надсилається запит до зовнішнього API (через `yahoo-finance2` бібліотеку), що дозволяє отримати актуальні фінансові показники [21].
- Розрахунок відсоткової зміни: На основі отриманих даних обчислюється відсоткова зміна ціни акції за тиждень (`weekly return`).
- Повернення відсортованого масиву: Результати обробки повертаються клієнту у вигляді відсортованого масиву об'єктів у форматі JSON [23].

```

7  const getCompanies = async (req, res) => {
9      const rawData = fs.readFileSync(companiesPath, "utf-8");
10     const companies = JSON.parse(rawData);
11
12     const results = await Promise.all(
13         companies.map(async (company) => {
14             try {
15                 const data = await fetchCompanyData(company.ticker);
16
17                 if (!data ||
18                     typeof data.marketCap !== "number" ||
19                     typeof data.stockPrice !== "number" ||
20                     typeof data.previousClose !== "number"
21                 ) {
22                     // ...
23                 }
24
25                 const weeklyReturn =
26                     ((data.stockPrice - data.previousClose) / data.previousClose) * 100;
27
28                 return {
29                     name: company.name,
30                     ticker: company.ticker,
31                     country: company.country,
32                     sector: company.sector,
33                     marketCap: data.marketCap,
34                     stockPrice: data.stockPrice,
35                     weeklyReturn: Number(weeklyReturn.toFixed(2)),
36                 };
37             }
38         });

```

Рис 3.4 – Фрагмент коду `companyController.js`

Завдяки такому підходу, API надає актуальні показники компаній майже в реальному часі, що є критично важливим для фінансового аналізу [1, 2].

```

{
  "name": "China Mobile",
  "ticker": "0941.HK",
  "country": "China",
  "sector": "Communication",
  "marketCap": 1952920502272,
  "stockPrice": 89.1,
  "weeklyReturn": 1.19
},

```

Рис 3.5 – Відповідь API у форматі JSON

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		31

Ця реалізація забезпечує повну та достовірну картину про ринки Азії [9].

Контролер `macroController.js` реалізує маршрут `/api/macros` і відповідає за отримання макроекономічних показників для визначеної країни. Назва країни передається як параметр запиту (`?country=Назва`), який потім внутрішньо конвертується у відповідний дволітерний ISO-код (наприклад, JP, CN, IN тощо) [5].

Для збору даних використовується World Bank API [5], з якого одночасно запитуються три основні індикатори:

- GDP — Валовий внутрішній продукт;
- Population — Населення країни;
- Unemployment — Рівень безробіття.

Контролер генерує паралельні запити для кожного з цих індикаторів, після чого витягує найбільш актуальне (перше доступне) значення для кожного з них. Для цього використовується допоміжна функція `findFirstValidValue`, яка перевіряє валідність та наявність даних у відповіді. У результаті клієнт отримує JSON-об'єкт з такими полями:

```
{  
  "country": "Japan",  
  "code": "JP",  
  "gdp": 4204494802431.55,  
  "population": 124516650,  
  "unemployment": 2.563  
}
```

Рис 3.6 – Відповідь API `/api/macros` у форматі JSON

Цей контролер є важливою складовою для порівняльного аналізу економічних показників між країнами, що відображається на відповідних сторінках клієнтської частини [1, 7]. Він забезпечує простий та стандартизований спосіб інтеграції відкритих міжнародних даних у вебзастосунок [5]

### 3.1.4 Конвертація валют

Сервіс `currencyService` розроблений для уніфікації фінансових показників компаній шляхом їх конвертації у долари США. У межах застосунку реалізовано окремий модуль, який містить фіксовані курси обміну основних азійських валют до USD. Всередині файлу визначено об'єкт `currencyToUSD`, що зберігає коефіцієнти конвертації: наприклад, для японської єни (JPY) курс встановлено як 0.006934 (тобто 1 єна = 0.006934 долара), для китайського юаня (CNY) – 0.1391, для індійської рупії (INR) – 0.01164, та інші. Цей словник включає курси для 10 різних валют, а також базову валюту USD з коефіцієнтом 1. Такий підхід дозволяє зберігати всі необхідні значення в пам'яті, усуваючи потребу в частих зверненнях до сторонніх API при кожній конвертації; фактично, обмінні курси ефективно кешуються як константи в самому застосунку.

```
const currencyToUSD = {
  JPY: 0.006934,
  CNY: 0.1391,
  HKD: 0.1274,
  INR: 0.01164,
  SGD: 0.7755,
  KRW: 0.0007,
  IDR: 0.000612,
  MYR: 0.2353,
  TWD: 0.0334,
  SAR: 0.266667,
  THB: 0.0308,
  USD: 1
};

function convertToUSD(value, currency) {
  const rate = currencyToUSD[currency];
  if (rate == null) {
    console.warn(`[currencyService] Невідомий курс для валюти "${currency}"`);
    return value;
  }
  return value * rate;
}

module.exports = {
  convertToUSD
};
```

Рис 3.7 – Реалізація сервісу конвертації валют

Основною функцією модуля є `convertToUSD(value, currency)`. Вона приймає числове значення та код валюти, після чого здійснює пошук відповідного курсу в об'єкті `currencyToUSD`. Якщо курс для вказаного коду валюти доступний, функція множить вхідне значення на знайдений

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		33

коефіцієнт і повертає результуючу суму вже у доларах США. У випадку, якщо валюта невідома (тобто відсутній ключ у довіднику), передбачено виведення попереджувального повідомлення в консоль, а функція повертає вихідне значення без змін. Приклад використання цього сервісу можна знайти у модулі Yahoo Finance (див. підрозділ 3.1.5), де дані про ціну акції та капіталізацію конвертуються за допомогою `convertToUSD` перед їх збереженням. Такий підхід забезпечує використання єдиної валюти для всіх розрахунків по компаніях. Код сервісу є досить лаконічним і продемонстрований у фрагменті вище, де показано структуру словника курсів та реалізацію функції конвертації.

### 3.1.5 Фінансові дані

Для отримання фінансових даних про компанії та їх подальшої уніфікації використовується модуль `yahooService`, який взаємодіє з Yahoo Finance API [21]. Цей сервіс інкапсулює логіку запиту ключових метрик компанії (ціна акції, ринкова капіталізація, фінансові показники тощо) і забезпечує кешування результатів для зменшення кількості повторних звернень до зовнішнього API. На початку файлу підключається стороння бібліотека `yahoo-finance2` [21] та файл `currencyService` для доступу до функції конвертації валют. Також завантажується існуючий кеш із локального файлу (`cache.json`) у змінну `cache`. Якщо файл із кешем вже існує, дані з нього розбираються як JSON – це дозволяє зберігати результати попередніх запитів між запусками сервера [23].

Основна експортована функція сервісу – `fetchCompanyData(ticker)`, яка повертає актуальні дані для компанії із заданим тікером. Логіка роботи така: спочатку перевіряється, чи присутній тікер у пам'яті кешу; якщо так, то негайно повертаються закешовані дані без здійснення нового мережевого запиту. В іншому випадку виконується запит до Yahoo Finance

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		34

API [21] методом quoteSummary з потрібними модулями даних.

```
21 const fetchCompanyData = async (ticker) => {
22   if (cache[ticker]) {
23     return cache[ticker];
24   }
25   try {
26     const quote = await yahooFinance.quoteSummary(ticker, {
27       modules: ["price", "summaryDetail", "financialData"],
28     });
29
30     if (!quote || !quote.price) return null;
31
32     const price = quote.price;
33     const finance = quote.financialData || {};
34     const currency = price.currency || "USD";
35
36     if (
37       typeof price.regularMarketPrice !== "number" ||
38       typeof price.regularMarketPreviousClose !== "number" ||
39       typeof price.marketCap !== "number"
40     ) return null;
41
42     const stockPriceUSD = convertToUSD(price.regularMarketPrice, currency);
43     const marketCapUSD = convertToUSD(price.marketCap, currency);
44
```

Рис 3.8 – Логіка функції запити даних з Yahoo Finance

Отриманий результат перевіряється на наявність основних полів, після чого відбувається конвертація валюти за допомогою currencyService [23].

### 3.1.6 Кешування та структура файлу realCompanies.json

Для початкового заповнення кешу даних застосовується окремий скрипт fetchAllToCache.js. Він одноразово завантажує інформацію про всі компанії азійського регіону зі стороннього API та зберігає результати у кеші на диску, щоб сервер міг надалі працювати з ними без затримок. Скрипт імпортує функцію конвертації валют з файлу currencyService.js та масив даних компаній із файлу realCompanies.json [23]. Цей JSON-файл містить базовий список компаній (назва, тікер, країна та сектор), які підлягають аналізу [23]. Структура даних у realCompanies.json являє собою масив об'єктів, де кожен об'єкт відповідає одній компанії. Наприклад, для

компанії можуть бути задані поля:

```
{ "name": "Mitsubishi UFJ Financial Group", "ticker": "8306.T", "country": "Japan", "sector": "Finance" },  
{ "name": "Sumitomo Mitsui Trust Holdings", "ticker": "8309.T", "country": "Japan", "sector": "Finance" },  
{ "name": "Sony Group", "ticker": "6758.T", "country": "Japan", "sector": "Technology" },
```

Рис 3.9 – Приклад полів для компанії

Такий підхід дозволяє гнучко додавати або змінювати перелік компаній, редагуючи лише зовнішній JSON-файл, без необхідності зміни коду логіки. Під час виконання скрипту всі ці дані завантажуються у змінну `companies` та ітеруються у циклі.

```
const cachePath = path.join(__dirname, "../data/cache.json");  
  
(async () => {  
  const cache = {};  
  
  for (const company of companies) {  
    const ticker = company.ticker;  
    try {  
      const quote = await yahooFinance.quoteSummary(ticker, {  
        modules: ["price", "summaryDetail", "financialData"],  
      });  
  
      const price = quote?.price || {};  
      const currency = price.currency || "USD";  
  
      cache[ticker] = {  
        ticker,  
        stockPrice: convertToUSD(price.regularMarketPrice ?? 0, currency),  
        marketCap: convertToUSD(price.marketCap ?? 0, currency),  
        previousClose: convertToUSD(price.regularMarketPreviousClose ?? 0, currency),  
        currency,  
      };  
      console.log(`[+] ${ticker} - OK`);  
    } catch {  
      cache[ticker] = null;  
      console.log(`[x] ${ticker} - error`);  
    }  
  }  
})
```

Рис 3.10 – Фрагмент скрипту `fetchAllToCache.js`

У циклі проходження по масиву компаній для кожного елемента береться біржовий тикер та виконується запит `quoteSummary` до Yahoo Finance [21]. Отримана відповідь обробляється: виділяється об'єкт `price` та визначається валюта котирувань. Далі для побудови запису кешу використовується функція `convertToUSD`: з її допомогою обчислюється значення полів `stockPrice`, `marketCap` і `previousClose` у доларах США [23]. Кожна успішно отримана інформація додається до об'єкта `cache` з ключем,

що дорівнює тікеру компанії. По завершенні роботи скрипт виводить повідомлення "Готово!" і примусово завершує процес.

### 3.2 Реалізація клієнтської частини

Файл `App.jsx` є визначальним елементом, що формує основну структуру клієнтського застосунку та відповідає за маршрутизацію між різними сторінками [16]. Цей компонент слугує своєрідною "точкою входу" для всіх інших елементів інтерфейсу. При монтуванні компонента `App` виконується перевірка готовності серверної частини шляхом надсилання спеціального запиту `health-check` до `/api/health` (для цього використовується HTTP-клієнт `Axios`). Доки сервер не поверне успішну відповідь, на екрані відображається компонент-завантажувач (`Loader`), і основний інтерфейс не рендериться. Такий підхід гарантує, що користувач побачить дані лише після того, як бекенд буде повністю готовий до роботи та надішле необхідну інформацію. Після успішного проходження цієї перевірки `App` відмалює загальний каркас застосунку, який включає: шапку сторінки (`<Header />`), підшапку з меню (`<Subheader />`), основний контент, що динамічно змінюється відповідно до поточного маршруту, а також нижній колонтитул (`<Footer />`).

Маршрути в застосунку задаються за допомогою компонента `<Routes>` з набором вкладених `<Route>` для кожного шляху [16]. Застосовано підхід `nested routing`, при якому окремі сторінки підключені як `React`-компоненти [17]. Зокрема, кореневий шлях `/` відображає компонент `Home` (головна сторінка), а шлях `/research` – компонент `Research` (сторінка аналітичних матеріалів). Для групи маршрутів, що відповідають секторам економіки, передбачено префікс `/section/`: наприклад, шлях `/section/finance` завантажує сторінку сектора "Finance", а `/section/technology` – сторінку сектора "Technology" тощо.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ док.ум.	Підпис	Дата		37

```

67 <Routes>
68   <Route path="/" element={<Home />} />
69   <Route path="/research" element={<Research />} />
70
71   <Route path="/sections" element={<Sections />} />
72   <Route path="/section/finance" element={<Finance />} />
73   <Route path="/section/technology" element={<Technology />} />
74   <Route path="/section/energy" element={<Energy />} />
75   <Route path="/section/health" element={<Health />} />
76   <Route path="/section/realestate" element={<RealEstate />} />
77   <Route path="/section/utilities" element={<Utilities />} />
78   <Route path="/section/industrials" element={<Industrials />} />
79   <Route path="/section/consumer" element={<Consumer />} />
80   <Route path="/section/materials" element={<Materials />} />
81   <Route path="/section/communication" element={<Communication />} />
82
83   <Route path="/countries" element={<Countries />} />
84   <Route path="/country/japan" element={<Japan />} />
85   <Route path="/country/china" element={<China />} />
86   <Route path="/country/india" element={<India />} />
87   <Route path="/country/thailand" element={<Thailand />} />
88   <Route path="/country/singapore" element={<Singapore />} />
89   <Route path="/country/indonesia" element={<Indonesia />} />
90   <Route path="/country/malaysia" element={<Malaysia />} />
91   <Route path="/country/taiwan" element={<Taiwan />} />
92   <Route path="/country/southkorea" element={<SouthKorea />} />
93   <Route path="/country/saudiArabia" element={<SaudiArabia />} />
94 </Routes>

```

Рис 3.11 – Структура компоненту App.jsx

Аналогічно, для сторінок окремих країн використано префікс /country/ – наприклад, /country/japan відкриває компонент сторінки "Japan", /country/india – сторінку "India" тощо. Також передбачені зручні узагальнені маршрути: /sections для огляду всіх секторів та /countries для огляду всіх країн. Усі ці сторінки імпортуються на початку файлу App.jsx і реєструються в маршрутизаторі. Нижче наведено фрагмент компонента App, де відображено підключення Header, Subheader, а також визначення основних маршрутів застосунку.

### 3.2.1 Домашня сторінка

Головна сторінка застосунку, реалізована в компоненті Home.jsx, призначена для відображення узагальненої інформації про азійський фондовий ринок [1]. При завантаженні Home виконується кілька запитів до API сервера, щоб отримати ключові дані: список усіх компаній, агреговані дані по секторах, по країнах, а також спеціальні добірки "лідерів" ринку. Ці

запити повертають вже підготовлені на сервері масиви об'єктів (наприклад, /api/top-movers – топ-5 компаній з найбільшим тижневим ростом, /api/sector-leaders – провідні компанії в кожному секторі) [1, 7].

Отримані дані зберігаються у стані компонента за допомогою хуків useState – для кожної категорії створено окреме поле стану (companies, sectors, countries, topMovers, sectorLeaders). Далі за допомогою методів масивів розраховуються узагальнені метрики: сумарна ринкова капіталізація всіх компаній (totalMarketCap), середня зміна за тиждень (avgWeeklyReturn), а також визначаються лідируючі сектор і країна за вкладом у ринок (тобто сектор з найбільшою часткою капіталізації та країна з найбільшим сукупним капіталом). Ці величини відображаються у верхній частині сторінки у вигляді окремих карток. Зокрема, реалізовано чотири інформаційні блоки: Total Market Cap (загальна капіталізація ринку, в трильйонах USD), Avg Weekly Return (середня відсоткова зміна за тиждень по всіх компаніях, з кольоровим індикатором прибутковості або збитковості), Top Sector (назва сектору-лідера та його частка у %), Top Country (назва країни-лідера та її сукупна капіталізація в млрд USD) [1, 2]. Таке подання дає користувачу узагальнену картину стану ринку на поточний момент.

```
5 const Home = () => {
6   const [companies, setCompanies] = useState([]);
7   const [sectors, setSectors] = useState([]);
8   const [countries, setCountries] = useState([]);
9   const [topMovers, setTopMovers] = useState([]);
10  const [sectorLeaders, setSectorLeaders] = useState([]);
11
12  useEffect(() => {
13    axios.get("/api/companies").then(res => setCompanies(res.data));
14    axios.get("/api/sectors").then(res => setSectors(res.data));
15    axios.get("/api/countries").then(res => setCountries(res.data));
16    axios.get("/api/top-movers").then(res => setTopMovers(res.data));
17    axios.get("/api/sector-leaders").then(res => setSectorLeaders(res.data));
18  }, []);
19
20
21  const totalMarketCap = companies.reduce((sum, c) => sum + (c.marketCap || 0), 0);
22  const avgWeeklyReturn = companies.length
23    ? companies.reduce((sum, c) => sum + (c.weeklyReturn || 0), 0) / companies.length
24    : 0;
```

Рис 3.12 – Запити до API серверу

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		39

Центральним елементом головної сторінки є інтерактивна карта Азії – компонент AsiaMap.jsx, що візуалізує географічний розподіл ринку [1, 9]. Карта реалізована на основі бібліотеки react-simple-maps, яка оперує гео-даними (GeoJSON). У нашому випадку використано заздалегідь підготовлений файл asia10.json, що містить контури країн Азійського регіону. Компонент AsiaMap завантажує з бекенду зведені дані по країнах (/api/countries) і будує словник capByCountry, де ключ – назва країни, а значення – її сумарна капіталізація (показник totalCap).

```
<Geography
  key={geo.rsmKey}
  geography={geo}
  children={<title>{countryName}</title>}
  onClick={() => {
    if (cap) {
      const slug = countryName.toLowerCase().replace(/ /, "");
      navigate(`/country/${slug}`);
    }
  }}
  style={{
    default: {
      fill: fill,
      outline: "none",
      opacity: cap ? 1 : 0.4,
      cursor: cap ? "pointer" : "not-allowed",
    },
    hover: {
      fill: "#2563eb",
      opacity: 1,
      outline: "none",
    },
    pressed: {
      fill: "#1d4ed8",
      outline: "none",
    },
  }},
```

Рис 3.13 – Обробка кліку на країні

При наведенні курсору на країну з'являється підказка з її назвою (реалізовано через SVG <title>), а при натисканні на країну, що має дані, здійснюється навігація на сторінку відповідної країни. Зокрема, при кліку формується строковий "slug" (наприклад, "South Korea" перетворюється на "southkorea") і викликається функція Maps() зі шляхом /country/..., яка за допомогою React Router відкриває сторінку країни [16]. Якщо ж інформація про країну не завантажена (в словнику capByCountry її капіталізація відсутня або нульова), обробник onClick ігнорує такі натискання, змінюючи курсор миші на недоступний (стиль not-allowed). Таким чином, інтерактивна карта дозволяє швидко переходити до

детального огляду конкретної країни, забезпечуючи інтуїтивну навігацію. Вище наведено фрагмент коду компонента AsiaMap.jsx, де видно реалізацію обробки події кліку на країні та умови доступності цього кліку.

### 3.2.2 Навігація між сторінками

Верхня навігаційна панель реалізована у компоненті Header.jsx. Вона відповідає за глобальну навігацію та містить назву застосунку, а також меню, що дозволяє переключатися між різними секціями та країнами. Дизайн компонента передбачає адаптивність: на малих екранах використовується "гамбургер"-меню, яке відкриває бічну панель (sidebar) з посиланнями, тоді як на великих екранах основні елементи можуть бути видимі у шапці [10, 11]. Логіка роботи Header побудована на двох станах: isSidebarOpen (булевий прапорець, що контролює відкриття/закриття бокового меню) та openSubmenu (зберігає назву відкритого підменю або null, якщо підменю закриті).

```
{isSidebarOpen && (  
  <div className="fixed top-0 left-0 h-full w-64 bg-white shadow-lg z-50 pt-16 overflow-y-auto">  
    <nav className="text-black text-base font-medium px-4 space-y-2">  
      <Link  
        to="/"  
        className="flex items-center space-x-2 hover:bg-gray-100 px-2 py-2 rounded"  
        onClick={() => setIsSidebarOpen(false)}  
      >  
        <Home className="w-5 h-5" />  
        <span>Home</span>  
      </Link>  
    </nav>  
  </div>  
)
```

Рис 3.14 – Випадаюче меню секторів

При натисканні на кнопку з іконкою меню (наприклад, SVG-іконка Menu у компоненті) стан isSidebarOpen перемикається, активуючи відображення або приховання бокової панелі. В середині бокової панелі розміщено навігаційний блок <nav> з посиланнями: перше посилання веде на головну сторінку (Home), а нижче організовано два випадаючі розділи –

Section (сектори) та Country (країни). Кожен з цих пунктів представлений кнопкою, яка при натисканні викликає функцію `toggleSubmenu(...)` з передачею імені відповідного меню. Ця функція встановлює значення `openSubmenu` рівним ідентифікатору меню (або прибирає його, якщо той уже відкритий), таким чином керуючи відображенням підменю.

На додаток до глобального заголовка, інтерфейс містить так звану підшапку – компонент `Subheader.jsx`, розташований безпосередньо під основним `Header`. Він надає користувачу засоби навігації для перегляду даних за країнами або за секторами, виконуючи роль панелі вкладок між двома режимами. Візуально `Subheader` складається з горизонтального меню, по центру якого знаходяться два основні пункти: `Section` і `Country`. Ці пункти не ведуть одразу на сторінки, а при наведенні курсору миші розкривають випадаючі списки (`dropdown`) з переліком відповідних підкатегорій.

```
<li
  className="relative cursor-pointer"
  onMouseEnter={() => handleMouseEnter("country")}
  onMouseLeave={handleMouseLeave}
>
  Country
  {hoveredMenu === "country" && (
    <div className="absolute bg-white border mt-2 shadow-md text-sm p-2 w-80 z-50 flex">
      <div className="w-full flex flex-col">
        <Link
          to="/countries"
          className="font-semibold px-2 py-1 mb-1 rounded hover:bg-gray-100 text-blue-600"
        >
          All countries
        </Link>
      </div>
      <ul className="flex flex-col flex-1 pr-2">
        {countryCol1.map((c) => (
          <li key={c} className="hover:bg-gray-100 px-2 py-1 cursor-pointer capitalize">
            <Link to={`/country/${c}`} className="block w-full h-full">
              {c}
            </Link>
          </li>
        ))}
      </ul>
    )}
  </div>
  </li>
```

Рис 3.15 – Випадаюче меню з переліком країн

Реалізація виконана через обробку подій `onMouseEnter` та `onMouseLeave` для елементів меню. Компонент використовує стан `hoveredMenu` (керується через `useState` та допоміжний `useRef` для тайм-

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		42

ауту), який зберігає назву меню, над яким зараз знаходиться курсор. При наведенні на пункт "Section" або "Country" викликається функція `handleMouseEnter(menu)`, що встановлює `hoveredMenu = "section"` чи `"country"` відповідно. При прибиранні курсору `handleMouseLeave` запускає невеликий тайм-аут (400 мс), після якого закриває випадаюче меню, якщо курсор не повернувся. Така невелика затримка при закритті покращує зручність використання (юзабіліті), дозволяючи користувачу перемістити курсор у межах випадаючого списку, не закриваючи його миттєво [10].

### 3.2.3 Сторінки окремих країн

Для відображення детальної інформації про кожну країну створено універсальний компонент `CountryPage.jsx`, який параметризується назвою країни. Цей компонент інкапсулює всю логіку завантаження та показу даних, що є спільною для сторінок типу "By country". Щоб уникнути дублювання коду для 10 країн, використано наступний підхід: існує один компонент `CountryPage`, що приймає через пропси назву країни (`country`), а для кожної конкретної країни створено невеликий компонент-обгортку (наприклад, файл `China.jsx`, `Japan.jsx` тощо), який просто імпортує `CountryPage` і викликає його зі своїм параметром. Таким чином, сторінка `China` містить виклик `<CountryPage country="China" />`, сторінка `Japan` – `<CountryPage country="Japan" />` і т.д. Така структура згадана при підключенні маршрутів у `App.jsx` [16]. В результаті, при переході на шлях `/country/china` React Router відрендерить компонент `China`, який фактично відобразить вміст `CountryPage` для `China`. Це сприяє підтримуваності: вся логіка зосереджена в одному місці, і додавання нової країни зводиться до запису її у конфігурацію (список країн) та створення одного файл-компонента з передачею параметра.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		43

```

const CountryPage = ({ country }) => {
  const [companies, setCompanies] = useState([]);
  const [macro, setMacro] = useState(null);

  useEffect(() => {
    axios.get("/api/companies")
      .then(res => {
        const filtered = res.data.filter(c => c.country === country);
        setCompanies(filtered);
      });

    axios.get(`/api/macros?country=${country}`)
      .then(res => setMacro(res.data))
      .catch(() => setMacro(null));
  }, [country]);

  const maxWeeklyReturn =
    companies.length > 0
      ? Math.max(...companies.map(c => c.weeklyReturn ?? -Infinity))
      : null;

```

Рис 3.16 – Завантаження даних у компоненті CountryPage.jsx

Компонент CountryPage при завантаженні виконує два запити до API: перший – для отримання списку компаній [1], другий – для отримання макроекономічних показників обраної країни [5]. У запиті /api/companies повертається повний перелік компаній, тому на стороні клієнта здійснюється фільтрація за полем country – відбираються тільки ті об’єкти, що відповідають цільовій країні. Такий підхід (отримати всі та відфільтрувати) є прийнятним з огляду на відносно невеликий обсяг списку (кілька десятків компаній). Другий запит – /api/macros?country=CountryName – повертає структуровані макропоказники для країни (наприклад, ВВП, населення, рівень безробіття) [5]. Результати обох запитів зберігаються у стані компоненту: масив відібраних companies та об’єкт macro з макроекономічними даними. Код реалізовано з використанням хука useEffect, який спрацьовує при зміні пропса country – тобто, якщо користувач перейде на сторінку іншої країни, компонент перевантажиться і виконає нові запити для оновленої країни. Фрагмент коду нижче демонструє цю логіку завантаження даних у CountryPage: спочатку з масиву усіх компаній відбираються тільки

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		44

компанії поточної країни, потім отримуються макро-індикатори для неї.

Після завантаження даних компонент формує структуру відображення. Назва країни виводиться як заголовок сторінки (`<h1>`), під яким розміщено блок макроекономічних показників. Ці показники беруться з об'єкту `macro`: наприклад, `GDP ($)` – валовий внутрішній продукт у млрд доларів [5], `Population` – населення (в млн осіб) [5], `Unemployment (%)` – рівень безробіття у відсотках [5]. В компоненті визначено словник `macroNames`, що задає відображувані назви та спливаючі підказки для кожного показника (джерело даних, рік тощо). Якщо дані `macro` успішно завантажено, компонент генерує для кожного показника окремий підблок: назва (із підказкою) та відформатоване значення або прочерк "-" якщо показник недоступний. Потім іде таблиця з переліком компаній цієї країни. Таблиця містить колонки: `Company` (назва компанії), `Sector` (сектор економіки), `Market Cap ($)` (ринкова капіталізація в доларах), `Stock Price` (поточна ціна акції) та `Weekly Return (%)` (зміна ціни за останній тиждень у відсотках) [1, 2]. Для заголовків колонок використовуються підказки `InfoTooltip` із поясненням фінансових термінів (визначені в об'єкті `tableTooltips`). Список компаній зберігається у стані `companies` – це масив об'єктів, кожен з яких вже містить поля з розрахованими на сервері даними (конвертованими у USD). Табличні рядки генеруються методом `.map()` по цьому масиву; для кращої читабельності рядки чергуються білим та сірим фоном (`idx % 2 === 0 ? "bg-white" : "bg-gray-50"`). Особливу увагу приділено колонці `Weekly Return`: окрім числового значення у відсотках, поруч із ним додається міні-компонент `WeeklyReturnIndicator` (стрілочка вгору або вниз відповідного кольору) та умовно значок "кубка" для найуспішнішої компанії (що має найбільший `weekly return`). Значок лідера визначається шляхом пошуку максимального значення `weeklyReturn` серед компаній (розраховується змінна `maxWeeklyReturn` під час рендерингу). Внизу сторінки передбачено

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		45

невеликий графік – гістограма розподілу щотижневих змін акцій компаній даної країни (BarChart з бібліотеки Recharts [19]). На цьому графіку по осі X відкладаються компанії (їх назви), а по осі Y – значення % зміни, що дає візуальне уявлення, в якій компанії був найбільший приріст або падіння за тиждень [1, 7]. У коді це досягається передачею масиву companies до компоненту <BarChart> та використанням weeklyReturn як ключа даних. Таким чином, сторінка країни містить як макрорівень (загальні економічні індикатори), так і мікрорівень (список компаній і їх динаміка) [1]. Всі дані на ній формуються з використанням інформації, що була згенерована на сервері з файлу realCompanies.json та інших джерел (макродані, фінансові показники) [23, 5]. Використання узагальненого компонента CountryPage гарантує єдиний підхід до відображення та полегшує підтримку – будь-які зміни в логіці (наприклад, додавання нового показника) потрібно внести лише в одному місці, і вони автоматично відобразяться для всіх країн.

### 3.2.4 Сторінка Research.jsx

Окремим розділом вебзастосунку є сторінка Research, що містить зібрані аналітичні матеріали та дослідження по країнах Азійського регіону [1]. На відміну від попередніх сторінок, які відображають переважно числові показники, ця секція представлена у формі бібліотеки статей/звітів, згрупованих за країнами. Основною метою є надання користувачу додаткового контексту та якісної інформації, такої як огляди ринків, результати досліджень або економічна аналітика, що виходить за рамки суто біржових показників [1, 2].

Інтерфейс сторінки Research.jsx побудований таким чином, щоб користувач міг зручно переглядати матеріали як по всіх країнах одразу, так і вибірково по одній країні. У верхній частині розміщено панель фільтрів у вигляді набору кнопок: одна кнопка "All" та по одній кнопці для кожної з

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		46

10 країн (Japan, China, India, ... Saudi Arabia). Реалізовано стан `selectedCountry`, який за замовчуванням дорівнює "All". Відповідно, при натисканні на певну кнопку стан оновлюється на обрану країну, що призводить до перерендеру списку статей. Кнопка "All" дозволяє швидко повернутися до перегляду матеріалів по всіх країнах одразу. Для покращення візуального сприйняття на кнопках країн відображаються маленькі емодзі-прапори (через функцію `flagEmoji(country)`, що повертає символ прапора). Активна кнопка (для якої `selectedCountry === country`) виділяється окремим стилем: білий фон і чорний текст, тоді як неактивні мають темний фон і сірий текст. Таким чином, панель фільтрів дає зрозумілий сигнал, які матеріали зараз на екрані, і дозволяє переключатися між наборами в один клік.

```

"China": [
  {
    title: "Regulating Conglomerates in China: Evidence from an Energy Conservation Program",
    source: "American Economic Review",
    description: "Вивчення впливу енергозберігаючих програм на китайські конгломерати.",
    link: "https://www.aeaweb.org/articles?id=10.1257/aer.115.4.1136",
  },
  {
    title: "An Analysis of the Political Economy of China's Enterprise Conglomerates",
    source: "Google Scholar",
    description: "Аналіз політичної економії китайських конгломератів на прикладі реформи електроенергетики.",
    link: "https://scholar.google.com/citations?view_op=view_citation&hl=en&user=yh4DSMYAAAAJ&citation_for_view",
  },
  {
    title: "Mobile Internet Business Models in China: Vertical Hierarchies, Horizontal Conglomerates, or Busine",
    source: "Google Scholar",
    description: "Моделі мобільного інтернет-бізнесу та роль конгломератів у Китаї.",
    link: "https://scholar.google.com/citations?view_op=view_citation&hl=en&user=CJ3ra3YAAAAJ&citation_for_view",
  },
],

```

Рис 3.17 – Відображення дослідницьких матеріалів

Сторінка Research, таким чином, доповнює суто кількісну інформацію застосунку якісними даними. Використовуючи зібрану автором базу посилань на зовнішні джерела (у кодуванні явно зазначено джерела як Wikipedia, HBR, журнали тощо), користувач може перейти за відповідними лінками для детального ознайомлення з темою [1, 7].

## ВИСНОВКИ ДО РОЗДІЛУ 3

У третьому розділі було детально розглянуто реалізацію програмної частини проєкту, що відповідає за економічний аналіз азійського регіону. Серверна логіка була розроблена на базі Node.js з використанням фреймворку Express [12, 13], що забезпечило швидку обробку запитів та гнучку архітектуру. Завдяки чіткому розділенню функціоналу на контролери, маршрути та утиліти, вдалося досягти високого рівня модульності та потенційної масштабованості системи [10, 11]. Особливу увагу приділили інтеграції із зовнішніми API, такими як Yahoo Finance [21] та World Bank [5], що дозволило використовувати актуальні фінансові та макроекономічні дані. Впровадження системи кешування та попереднє завантаження інформації сприяло підвищенню продуктивності та надійності функціонування сервера.

Клієнтська частина застосунку реалізована з використанням бібліотеки React [16], що забезпечило динамічну візуалізацію інформації та інтуїтивно зрозумілий інтерфейс користувача. Архітектура компонентів дає можливість відображати дані як у розрізі окремих країн, так і за секторами економіки, з інтерактивною картою Азії та зведеною аналітикою на головній сторінці [1, 9]. Використання спеціалізованих компонентів для навігації та представлення інформації (таких як CountryPage, SectionPage, AsiaMap) дозволяє гнучко розширювати функціонал у майбутньому [10, 17].

Таким чином, у ході реалізації програмної частини було дотримано сучасних принципів розробки, що сприяло створенню стабільного, розширюваного та ефективного вебзастосунку для цілей економічного аналізу.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		48

## РОЗДІЛ 4. Дослідження та аналіз розробленої системи

### 4.1 Загальна оцінка працездатності системи

Розроблений програмний комплекс був інстальований та розгорнутий у локальному операційному середовищі з метою всебічного вивчення функціональності як клієнтської, так і серверної підсистем. Проведені випробування підтвердили успішний запуск компонентів користувацького інтерфейсу та серверного API, а також їхню ефективну взаємодію, що забезпечує коректне виконання всіх передбачених функцій. Під час тестових випробувань встановлено, що всі основні вебсторінки застосунку, зокрема "Home", "Country", "Section" та "Research", завантажуються без виявлення програмних помилок та демонструють актуальні дані. Цей результат свідчить про загальну працездатність системи та належну інтеграцію всіх її складових елементів.

Ключовим аспектом проекту є реалізація інтеграції із зовнішніми джерелами інформації. Зокрема, для отримання фінансових показників компаній, таких як поточні біржові котирування та показники ринкової капіталізації, було використано API Yahoo Finance [21]. Це забезпечило можливість автоматичного завантаження найновіших даних про компанії Азійського регіону, усуваючи необхідність ручного оновлення інформації. Окрім того, статистичні макроекономічні дані, включаючи валовий внутрішній продукт та чисельність населення, завантажуються через API Світового банку [5], який надає програмний доступ до значного обсягу глобальних індикаторів. Таким чином, система інтегрована з авторитетними інформаційними джерелами, що дозволяє їй оперативно оновлювати інформацію в режимі реального часу. Результати локального

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		49

розгортання та тестування переконливо підтвердили успішне встановлення зв'язку з обома API – Yahoo Finance [21] та World Bank [5]: зовнішні дані належним чином отримуються та коректно відображаються на відповідних вебсторінках застосунку.

## 4.2 Перевірка функціональності ключових сторінок

Була проведена детальна перевірка роботи основних інтерфейсних сторінок застосунку. Кожна сторінка протестована на предмет правильності відображення даних, інтерактивності елементів та відповідності заданій логіці.

### 4.2.1. Сторінка Home

Головна сторінка вебзастосунку є інтерактивною панеллю приладів, що відображає оглядові дані фондового ринку Азії [1]. На ній представлено декілька інформаційних блоків. Перший з них — інтерактивна карта Азії, яка демонструє основні країни регіону та їхні ринкові показники [9, 16].



Рис. 4.1 – Інтерактивна карта на головній сторінці

Вона дозволяє користувачу обрати країну (клікнувши на карту) для переходу до детальнішого перегляду її показників. Карта кольорово виділяє країни, присутні у базі даних застосунку, і містить підказки з назвою країни та сумарною ринковою капіталізацією. Такий візуальний огляд забезпечує швидке уявлення про охоплені ринки та їхній масштаб.

Під картою розміщено таблицю Top Movers, в якій наведено перелік топ-5 компаній регіону за тижневим приростом курсу акцій [1, 2]. Таблиця містить назву компанії, країну походження та відсоток зміни ціни за останній тиждень. На рисунку 4.2, що демонструє таблиці "Top Movers" та "Sector Leaders" на головній сторінці, видно, що значення прибутковості позначені кольором (зелений для зростання, червоний для падіння) для наочності. Ця таблиця дає змогу швидко ідентифікувати лідерів і аутсайдерів тижня на ринку.

Top Movers (Weekly Return)			Sector Leaders		
Company	Country	Weekly Return (%)	Sector	Company	Market Cap (\$)
ACWA Power	Saudi Arabia	6.72	Finance	China Construction Bank	235.7 млрд
YTL Power International	Malaysia	6.69	Technology	TSMC	864.4 млрд
POSCO Holdings	South Korea	4.68	Energy	Saudi Aramco	1599.5 млрд
POSCO	South Korea	4.68	Health	Sun Pharmaceutical	47.0 млрд
Yangzijiang Shipbuilding	Singapore	4.52	Real Estate	Mitsui Fudosan	26.0 млрд
			Consumer	Alibaba Group	287.6 млрд
			Industrials	Hitachi	127.5 млрд
			Communication	China Mobile	249.1 млрд
			Materials	China Shenhua Energy	103.9 млрд
			Utilities	China Yangtze Power	101.4 млрд

Рис. 4.2 – Таблиця “Top Movers” та “Sector Leaders”

Окрім того, головна сторінка відображає секцію Sector Leaders, яка показує компанії-лідери в кожному секторі економіки. У вигляді іншої таблиці представлено список секторів та відповідних найбільших компаній за ринковою капіталізацією [1, 2]. На рисунку 4.2, що ілюструє таблиці "Top Movers" та "Sector Leaders" на головній сторінці, для кожного сектора вказано назву провідної компанії та її капіталізацію. Таким чином, сторінка Home надає цілісний огляд стану ринку: через карту —

географічний розріз, а через дві таблиці — розріз по компаніях та секторах.

## 4.2.2 Сторінка CountryPage

Сторінка країни призначена для детального перегляду показників вибраної країни регіону. Для прикладу було перевірено сторінки двох найбільших економік — China (Китай) та Japan (Японія). На сторінці кожної країни у верхній частині відображається назва країни та основні макроекономічні показники [5]. Зокрема, показано значення валового внутрішнього продукту (GDP), чисельності населення та рівня безробіття. Ці дані отримуються через World Bank API [5] і наведені за актуальними доступними періодами (останній рік доступної статистики).

### China



Рис. 4.3 – Відображення макропоказників для Китаю

На сторінці CountryPage демонструється блок із трьох метрик: GDP (у млрд дол. США), Population (населення, млн осіб) та Unemployment (рівень безробіття, %) [5]. Для Японії аналогічно відображаються її макроекономічні параметри.

### Japan



Рис. 4.5 – Макроекономічні показники для Японії

Всі показники представлені у зрозумілому форматі з розрядністю чисел та супровідними підказками (tooltip) щодо одиниць виміру та джерела даних.

Наступним блоком на сторінці країни є таблиця компаній, зареєстрованих у цій країні і наявних у базі даних системи [1, 2]. У таблиці наведено перелік компаній із зазначенням їх сектора економіки, ринкової капіталізації, поточної ціни акції та тижневої зміни (%). Кожна друга стрічка підсвічена іншим відтінком фону для покращення читабельності. У стовпці Weekly Return відображаються значення тижневої динаміки, причому для найкращої компанії за цим показником додається спеціальний значок нагороди. Це допомагає швидко визначити лідера зростання в межах країни. Загалом перевірка показала, що таблиця коректно відображає всі компанії для обраної країни та динамічно оновлюється при виборі іншої країни (через зміну маршруту URL).

Company	Sector	Market Cap (\$) ?	Stock Price ?	Weekly Return (%) ?
China Mobile	Communication	244 326 437 014,733	11,148	0.91% ▲
JD.com	Consumer	47 922 925 155,123	16,524	-0.69% ▼
CNOOC Limited	Energy	115 605 981 193,83	2,352	0.87% ▲
China Construction Bank	Finance	234 924 021 723,955	0,927	-0.27% ▼
Sinopharm Group	Health	7 450 500 852,122	2,387	0.21% ▲
China State Construction Engineering	Industrials	32 704 228 119,347	0,791	0.53% ▲
China Shenhua Energy	Materials	106 085 523 691,93	4,312	1.80% ▲ 🏆
China Resources Land	Real Estate	23 756 796 520,038	3,332	-0.19% ▼
Tencent Holdings	Technology	594 447 951 095,398	65,356	-0.39% ▼
China Yangtze Power	Utilities	102 003 688 944,435	4,169	0.57% ▲

Рис. 4.6 – Табличне відображення компаній Китаю

У нижній частині сторінки CountryPage розміщена діаграма типу “стовпчаста діаграма”, що візуалізує показник Weekly Return (%) для компаній даної країни [7, 19]. На осі абсцис зазначені компанії, а висота стовпців відповідає відсотковій зміні їх акцій за тиждень. Рисунок 4.7 ілюструє, як діаграма дозволяє візуально порівняти динаміку різних компаній.

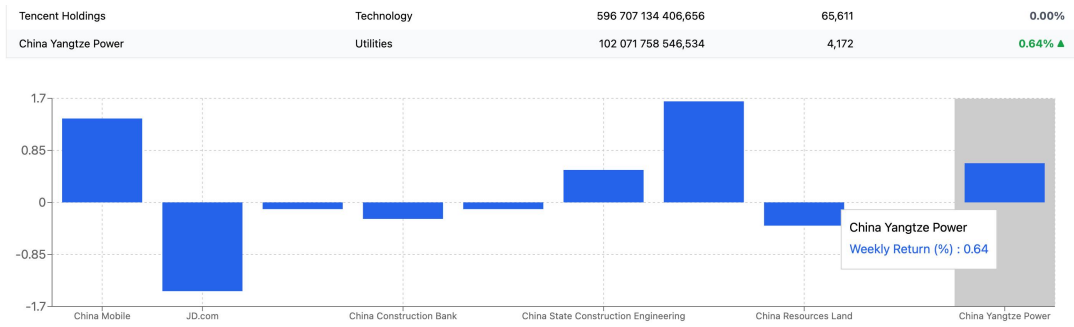


Рис. 4.7 – Графічне відображення тижневої дохідності акцій компаній Китаю

При наведенні курсору на стовпець відображається підказка з точним значенням відсотка. Перевірка підтвердила, що діаграма будується правильно на основі актуальних даних таблиці.

#### 4.2.3 Сторінка SectionPage

Для кожного сектора економіки в системі передбачена окрема сторінка, що відображає деталі цього сектора по всіх компаніях [1]. Перевіримо працездатність на прикладі сектору Utilities (Комунальні послуги). Сторінка сектора починається заголовком із назвою сектора, після чого виводиться короткий підсумок: зазначено лідера сектора – найбільшу за капіталізацією компанію в цьому секторі [1, 2]. Це значення обчислюється на основі даних усіх компаній сектору та відображається як текстова вставка (назва лідера і його капіталізація в доларах США). Такий акцент дозволяє користувачу одразу побачити ключового гравця у галузі. Нижче розташована розгорнута таблиця компаній, що належать до обраного сектора [1, 2]. Рисунок 4.8, що демонструє сторінку сектора Utilities зі списком компаній, показує приклад заповнення таблиці: стовпці містять назву компанії, країну її походження, ринкову капіталізацію, ціну акції та тижневу зміну курсу.

## Utilities

Лідер сектора: China Yangtze Power (Market Cap: 102 003 688 944,435 \$)

Company	Country	Market Cap (\$) ↑	Stock Price ↑	Weekly Return (%) ↑
China Yangtze Power	China	102 003 688 944,435	4,169	0.57% ▲
ACWA Power	Saudi Arabia	52 677 325 929,775	71,947	6.72% ▲ 📈
NTPC Limited	India	37 167 809 597,276	3,825	-0.18% ▼
Tenaga Nasional	Malaysia	19 449 240 635,802	3,337	-0.14% ▼
Korea Electric Power Corp	South Korea	13 189 150 828,134	20,545	-1.01% ▼
Kansai Electric Power	Japan	12 208 973 771,178	10,959	0.35% ▲
SP Group	Singapore	9 286 736 431,104	5,219	1.20% ▲
Electricity Generating Public Co	Thailand	1 735 018 070,016	3,296	-0.47% ▼
Cikarang Litrindo	Indonesia	663 621 080,187	0,042	0.00%
Taiwan Power Company	Taiwan	68 092 495,565	0,695	2.46% ▲

Рис. 4.8 – Сторінка сектора Utilities

Над кожним із заголовків стовпців передбачені підказки, що пояснюють економічний зміст показників (наприклад, “Market Cap” – це загальна ринкова вартість компанії, “Stock Price” – поточна ціна акції, “Weekly Return” – тижнева процентна зміна).

Як і на сторінці країни, на сторінці сектора реалізовано графічне відображення даних. Внизу розміщено стовпчасту діаграму, де по осі X відкладені назви компаній сектора, а по осі Y – їх ринкова капіталізація (у млрд доларів) [7, 19]. Для наочності кожен стовпець одного кольору (прийнято фіолетовий відтінок для всіх секторів). Діаграма масштабована під розмір контейнера і автоматично підлаштовується під довжину назв компаній.

Рисунок 4.9, що демонструє діаграму ринкової капіталізації компаній сектора (на прикладі сектора Utilities), ілюструє розподіл капіталізації між компаніями: чітко видно, яка компанія є найбільшою за цим показником у галузі.

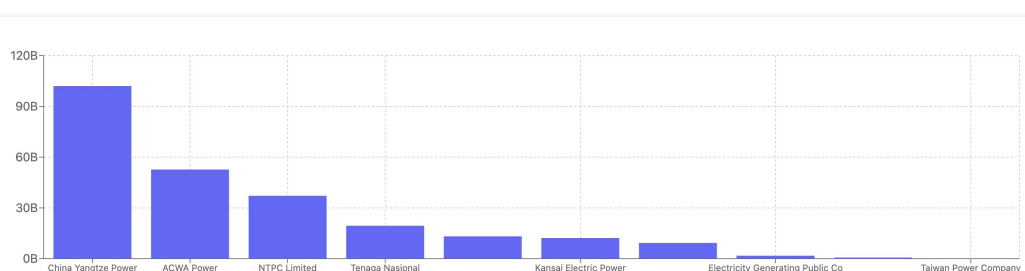


Рис. 4.9 – Діаграма ринкової капіталізації компаній сектора

Інтерактивність графіку проявляється через спливаючі підказки з точними значеннями при наведенні. Результати тестування підтвердили узгодженість цієї графічної інформації з табличними даними вище, що свідчить про правильну роботу компоненту діаграми.

#### 4.2.4 Сторінки огляду Countries і Sections

Для зручності навігації система містить окремі оглядові сторінки, що перераховують усі доступні країни та сектори [10]. Сторінка Countries (маршрут /countries) відображає список всіх країн Азії, які представлені в застосунку. Ця сторінка виконана у вигляді сітки карток (карточок), де кожна картка відповідає окремій країні [9]. Рисунок 4.10, що демонструє оглядову сторінку країн (Countries), показує приклад такого подання: назва країни подана як гіперпосилання, по кліку на яке відкривається сторінка відповідної країни [16].

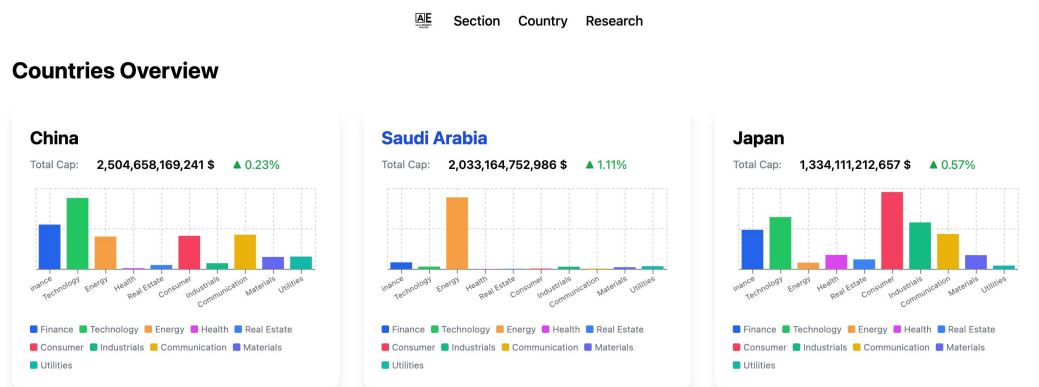


Рис. 4.10 – Оглядова сторінка країн

На картці також вказано сумарну ринкову капіталізацію всіх компаній цієї країни (позначено “Total Cap”) та середню тижневу зміну (у відсотках) по всіх її компаніях [1, 2]. Поруч з відсотковим значенням відображається стрілка вгору або вниз, яка інтуїтивно відображає позитивну чи негативну середню динаміку. Додатково, на кожній картці

країни розміщено міні-діаграму, що показує розподіл ринкової капіталізації між секторами всередині країни [7, 19]. Ця діаграма виконана у вигляді набору маленьких стовпців різних кольорів, кожен з яких відповідає певному сектору (використано 10 кольорів для 10 секторів). Таким чином, переглянувши картку країни, користувач може оцінити, які сектори домінують в її ринковій структурі. Перевірка працездатності показала, що всі картки країн правильно відображають дані (наприклад, для крупних економік, таких як Китай чи Японія, на картці видно відповідні великі значення Total Cap та значну частку Technology чи Finance секторів у їх міні-діаграмах), що узгоджується з очікуваними реальним тенденціям.

Аналогічно реалізована сторінка Sections (маршрут /sections), що перелічує всі сектори [10]. Рисунок 4.11, що ілюструє оглядову сторінку секторів (Sections), містить сітку карток по кожному сектору економіки.

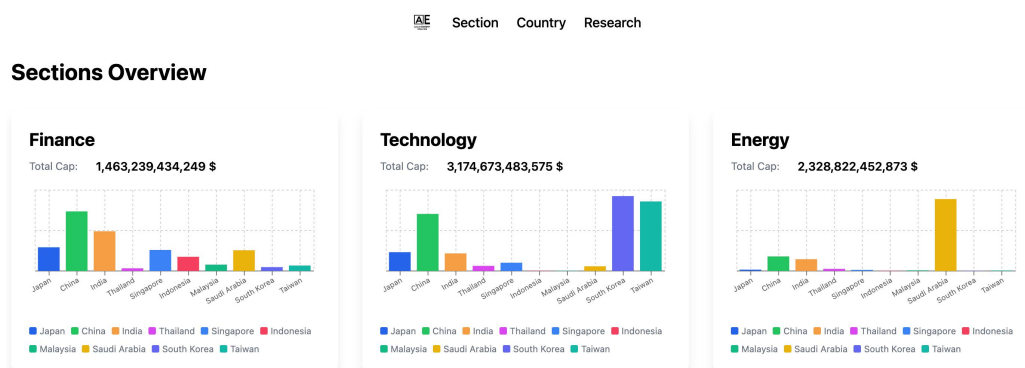


Рис. 4.11 – Оглядова сторінка секторів

На картці сектора зазначено назву сектора (із можливістю переходу на його сторінку) та наведено сумарну вагу цього сектора в ринку (Market Weight, у відсотках від загальної капіталізації ринку) [1, 2]. Також для сектора може бути вказано загальну ринкову капіталізацію всіх його компаній. Кожна картка сектора містить невеликий графічний елемент – діаграму розподілу капіталізації цього сектора за країнами [7, 19]. Це дає

уявлення, в яких країнах сектор найбільш розвинутий (за сумарною капіталізацією компаній). В ході тестування підтверджено, що ці картки коректно заповнені: для сектора Technology помітно переважання внеску таких країн як Японія та Китай, для сектора Energy – вагомий внесок країн Близького Сходу тощо, що відповідає очікуваним реальним тенденціям.

#### 4.2.5 Сторінка Research

Сторінка Research призначена для відображення аналітичної інформації та матеріалів дослідження ринків [1]. Її зміст структуровано за країнами: користувач може обрати перегляд аналітики All (усі країни) або вибрати конкретну країну за допомогою інтерактивного фільтра у вигляді набору кнопок. При виборі країни, на сторінці відображаються тільки матеріали, що стосуються відповідного ринку. Перевірка функціональності показала, що фільтр працює правильно – наприклад, при виборі “China” на сторінці залишаються лише аналітичні записи, пов’язані з Китаєм.

Основний зміст сторінки Research – це список аналітичних статей чи звітів по країнах. Матеріали згруповано за країною: назва країни подається підзаголовком, під яким перераховані один за одним аналітичні дописи. Кожен допис містить заголовок (тему дослідження чи статті), короткий опис та посилання на джерело (за наявності) [1, 7]. Для наочності кожен елемент списку промарковано порядковим номером і виділено рамкою. Рисунок 4.12, що ілюструє сторінку Research з аналітичними матеріалами, показує приклад наповнення для декількох країн: видно, що для кожної країни наведено 2–3 стислих огляди з вказівкою джерела (назви аналітичного порталу або видання).

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		58

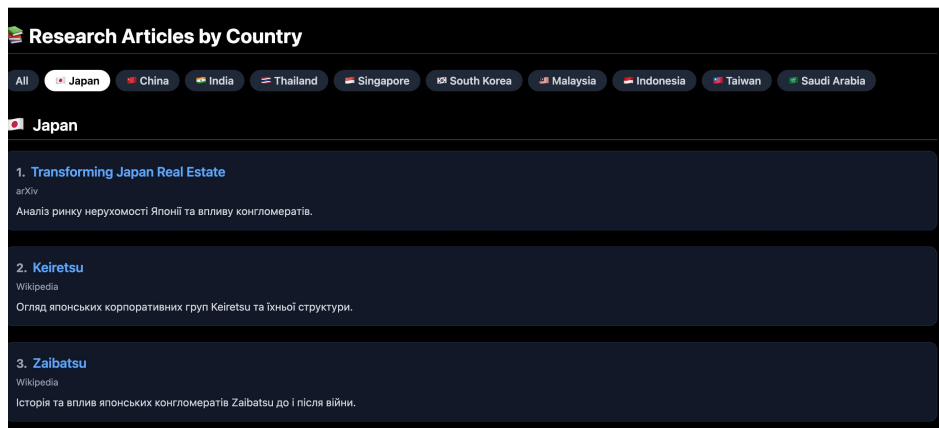


Рис. 4.12 – Сторінка Research

Ця інформація носить ознайомчий характер і може служити відправною точкою для глибшого аналізу ринків. В ході тестування верифіковано, що всі посилання працюють (відкривають відповідні зовнішні ресурси у новому вікні), а для країн, по яких ще немає даних, система відображає повідомлення про відсутність аналітичних матеріалів. Таким чином, сторінка Research функціонує стабільно, забезпечуючи коректну фільтрацію та відображення контенту, а також відповідає вимогам щодо інформаційного наповнення розділу аналітики.

#### 4.2.6 Навігація

Логіка навігації вебзастосунку реалізована через поєднання головного заголовка сторінки (Header) та додаткової панелі навігації (Subheader) [10]. У верхній частині інтерфейсу постійно знаходиться Header – фіксована шапка чорного кольору, яка містить кнопку меню (три лінії – “гамбургер”) і назву застосунку. Натискання на кнопку меню викликає бокову панель навігації (sidebar), що дозволяє користувачу швидко перейти до основних розділів [11]. В боковому меню представлені пункти: Home, Section, Country та Research. Пункти Section і Country мають випадаючі підменю: при натисканні на Section відображається список з

усіма секторами, а при натисканні на Country – список усіх країн. Це дає змогу в один клік обрати потрібну категорію даних.



Рис. 4.13 – Відображення Sidebar

Пункти меню оздоблені піктограмами для наочності (наприклад, піктограма будинку для Home, книжки для Research тощо). Після вибору розділу меню автоматично згортається, що зручно при навігації на мобільних пристроях або малих екранах.

Додатково реалізовано Subheader – горизонтальну панель під шапкою, яка містить меню верхнього рівня. В Subheader відображається логотип застосунку (клік по ньому завжди повертає на Home) та текстові пункти: Section, Country, Research. При наведенні курсору на пункт Section з’являється випадаюче вікно, розділене на дві колонки, де перелічені всі 10 секторів (а також пункт “All sections” для переходу на оглядову сторінку всіх секторів). Аналогічно при наведенні на пункт Country з’являється випадаючий список з 10 країнами та пунктом “All countries”.

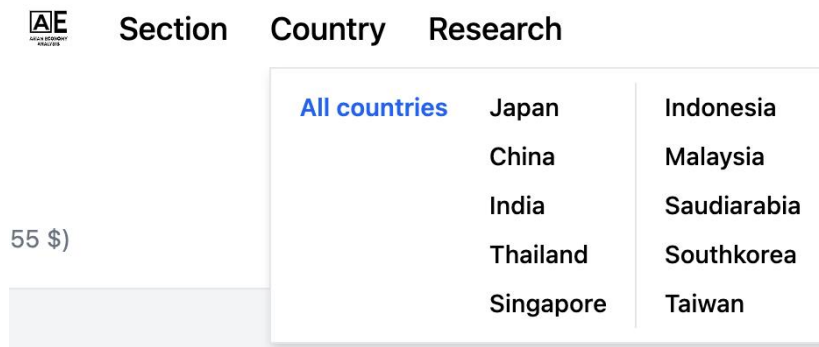


Рис. 4.14 – Навігація Subheader

Таким чином, Subheader дублює функціональність бокового меню, але в форматі, зручному для десктопного інтерфейсу. Навігаційна структура побудована логічно: користувач завжди має доступ як до загальних оглядів (через All countries/sections), так і до конкретних сторінок. Тестування підтвердило коректність роботи навігації: всі посилання ведуть на вірні сторінки, випадаючі меню відкриваються та закриваються належним чином, а активний розділ підсвічується для візуальної індикації. Така організація навігації відповідає принципам зручності користування і задовольняє вимоги технічного завдання щодо доступності всіх розділів системи.

### 4.3 Оцінка швидкодії та стабільності

Важливим етапом дослідження є оцінка продуктивності системи, зокрема швидкодії відображення даних та стабільності роботи при навантаженнях. Для забезпечення прийнятної швидкодії у вебзастосунку реалізовано механізм кешування даних [23]. Суть його полягає у попередньому збереженні найбільш важливих даних про компанії в локальному сховищі, щоб уникнути багаторазових звернень до зовнішніх API під час роботи користувача. Зокрема, було розроблено скрипт fetchAllToCache.js, який одноразово звертається до API Yahoo Finance [21]

для всіх компаній зі списку та зберігає отримані результати у файл кешу (формат JSON). Виконання цього скрипту забезпечує створення локального файлу cache.json з актуальними фінансовими показниками всіх компаній (ринкова капіталізація, ціна акцій тощо). Після цього серверна частина системи при обробці запитів спочатку перевіряє наявність даних у кеші і, за можливості, використовує їх [23]. Такий підхід значно зменшує кількість безпосередніх звернень до зовнішнього API під час роботи користувача і прискорює відповідь. Відомо, що основна мета кешування – зберігати дані, які можна швидко повернути при повторних запитах, знижуючи затримки та покращуючи час завантаження контенту [23]. У нашому випадку використання кешу дозволяє миттєво отримувати раніше завантажені дані про компанії, що підвищує швидкодію інтерфейсу при багаторазовому перегляді сторінок. Для ілюстрації роботи з API наведемо фрагмент функціоналу запиту даних однієї компанії. Функція fetchCompanyData(ticker) на сервері виконує такий алгоритм: спочатку перевіряється, чи є запис з даними по запитуваному тикеру в кеші; якщо є – повертається закешована інформація, якщо ні – здійснюється зовнішній запит до Yahoo Finance API [21].

Зовнішній запит реалізовано через бібліотеку Yahoo Finance (yahoo-finance2) методом quoteSummary, який повертає структуру даних про ціну акції, капіталізацію та інші фінансові метрики для заданого тикера [21]. Після отримання відповіді функція перевіряє наявність необхідних полів і конвертує отримані значення до єдиної валюти (USD) за потреби [23]. Потім зібрані дані зберігаються до кешу і повертаються викликачу. Таким чином, при першому запиті по новій компанії система трохи зачекає на відповідь Yahoo API, але всі наступні звернення будуть обслуговуватися вже з локального кешу, що відбувається майже миттєво. Перевірка цього механізму показала, що після початкового завантаження даних час відкриття сторінок значно скорочується: повторне відкриття сторінки

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		62

певної країни чи сектора відбувається без помітних затримок, оскільки необхідна інформація вже є в локальному сховищі.

Схожий підхід застосовано і для макроекономічних показників: сервер має можливість кешувати результати запитів до API Світового банку [5], аби не звертатися щоразу за одними й тими ж даними, які оновлюються відносно рідко. Це додатково підвищує стабільність роботи системи при відсутності інтернет-з'єднання або при перевищенні лімітів запитів до API.

Окрім швидкодії, було перевірено стабільність роботи системи при різних сценаріях використання. Застосунок демонструє стабільну роботу при багаторазовому перемиканні між сторінками, швидкому переході по різних розділах, одночасному відкритті декількох вкладок браузера тощо – жодних збоїв чи витоків пам'яті не виявлено. Інтерфейс лишається чутливим до дій користувача навіть при завантаженні великих обсягів даних, оскільки більшість важкої обробки виконується на стороні сервера та у фоновому режимі (під час кешування).

Загалом, реалізоване кешування даних і оптимізація запитів позитивно вплинули на продуктивність: система швидко завантажує інтерактивні компоненти (карти, діаграми) навіть при середній швидкості мережі, а використання локального кешу мінімізує навантаження на зовнішні служби. Це підтверджує відповідність проєкту вимогам щодо швидкодії та стабільної роботи під навантаженням.

#### **4.4 Аналіз відповідності технічним вимогам**

Підсумковим етапом роботи є зіставлення реалізованого функціоналу системи з початковими технічними вимогами проєкту. Згідно з технічним завданням, система повинна була забезпечувати ряд ключових можливостей, зокрема: інтерактивне відображення біржової інформації по

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		63

країнах Азії, наявність сторінок з детальними даними по кожній країні та сектору, реалізація засобів навігації між цими сторінками, а також відображення аналітичних матеріалів [1]. Аналіз показує, що всі заплановані функції успішно реалізовані у фінальній версії продукту.

По-перше, вимога щодо інтерактивної карти була виконана: на головній сторінці відображено карту Азії, інтегровану з навігацією (клік по країні переводить на сторінку детального огляду) [9, 16]. Карта охоплює усі цільові країни і відповідає критеріям інтерактивності (наведення відображає назву країни, доступність залежить від наявності даних).

По-друге, сторінки країн та секторів реалізовані повністю згідно з технічними вимогами: кожна з цих сторінок містить необхідні таблиці з переліком компаній та ключовими показниками, а також графічні діаграми для кращої наочності, як і передбачалося проєктом [1, 2, 7, 19]. Сторінки країн доповнені макроекономічними даними, що не лише відповідало вимогам, а й підвищило цінність застосунку з точки зору аналітики (даний пункт було включено до ТЗ і виконано за допомогою інтеграції з World Bank API) [5].

Навігаційна структура системи повністю відповідає запланованій [10, 11]: користувач має можливість переглядати загальні списки країн та секторів (реалізовано окремими оглядовими сторінками), а також швидко переходити до потрібної сторінки через меню. Наявність подвійного меню (“гамбургер”-меню для мобільних та горизонтального меню для десктопу) не суперечить вимогам, а навпаки – покращує юзабіліті, що було запропоновано під час розробки як удосконалення. Всі переходи між сторінками здійснюються без перезавантаження сторінки (SPA-навігація засобами React) [16], що скорочує час очікування та було передбачено архітектурою системи.

Функціонал розділу Research (аналітика ринку) також реалізовано у відповідності до поставлених цілей [1]. Зокрема, система надає

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ док.ум.	Підпис	Дата		64

користувачу додаткову інформацію (аналітичні статті та огляди) по кожній країні, що виходить за рамки суто числових фінансових показників. Це сприяє більш глибокому розумінню контексту ринків і було вказано як бажана функціональність у технічному завданні. Кожен аналітичний матеріал містить джерело, що відповідає вимогам достовірності інформації.

Варто зазначити, що всі компоненти інтерфейсу та логіки, описані в технічному завданні (ТЗ), знайшли своє відображення в готовому продукті. Наприклад, у ТЗ була передбачена функція визначення лідерів по сектору та країні – в системі це реалізовано у вигляді відповідних блоків (лідера сектора на сторінці сектора, топ-компанії тижня на головній сторінці тощо) [1, 2]. Інтеграція з Yahoo Finance API [21] і Світовим банком [5] повністю відповідає вимогам щодо автоматизації збору даних. Структура даних та формат представлення інформації на сторінках відповідають описаному у ТЗ прототипу: використано таблиці, графіки, картографічний модуль – згідно з проєктними рішеннями, розробленими у попередніх розділах [1, 2, 7, 9, 19].

Отже, проведений аналіз демонструє, що розроблена система повністю відповідає технічним вимогам і специфікаціям. Усі заплановані функціональні можливості реалізовано в повному обсязі, а деякі аспекти (як-от адаптивна навігація, детальні макропоказники) навіть перевищують базові вимоги, покращуючи якість продукту. Структура і поведінка системи узгоджуються із проєктною документацією, що свідчить про успішне виконання поставлених завдань дипломного проєкту.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		65

## ВИСНОВКИ ДО РОЗДІЛУ 4

У четвертому розділі було представлено результати тестування та дослідження функціонуючої системи. Проведена загальна оцінка працездатності підтвердила, що вебзастосунок успішно запускається локально, встановлює з'єднання з зовнішніми API [5, 21] та коректно відображає інформацію на всіх основних сторінках. Детальна перевірка функціональності ключових інтерфейсів (головна сторінка, сторінки країн, секторів, навігаційні сторінки та розділ аналітики) продемонструвала відповідність реалізації поставленим вимогам. Кожна сторінка забезпечує користувача необхідними інтерактивними елементами та даними: на карті відображено охоплення ринку, у таблицях – детальні фінансові показники, на графіках – наочну динаміку, а в розділі Research – контекстну аналітичну інформацію [1, 2, 7, 9, 19]. Система показала стабільну роботу під час перемикання між різними видами даних, підтвердивши надійність розробленої архітектури.

Також у цьому розділі проаналізовано показники продуктивності. Реалізація кешування даних [23] суттєво підвищила швидкодію: користувач отримує інформацію майже миттєво при повторних зверненнях до вже завантажених даних. Інтеграція з Yahoo Finance [21] та Світовим банком [5] не лише надала актуальні дані, але й була оптимізована з точки зору навантаження – через проміжне збереження результатів запитів. Це забезпечило стабільність системи та відсутність затримок навіть при інтенсивному використанні.

Зіставлення функціональних можливостей готового продукту із технічним завданням показало повну відповідність: всі заплановані модулі і компоненти успішно реалізовано, а їх робота відповідає описаним сценаріям використання. Таким чином, у розділі 4 експериментально підтверджено працездатність і ефективність розробленої інформаційної

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		66

системи, а також продемонстровано, що вона відповідає заданим критеріям якості та функціоналу.

					ІАЛІЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		67

## ВИСНОВКИ

У кваліфікаційній роботі було розроблено та досліджено вебзастосунок для аналізу фондового ринку Азійського регіону. В результаті виконання поставлених завдань створено інтерактивну інформаційну систему, що агрегує дані про основні публічні компанії десяти країн Азії, відображає їх фінансові показники та надає користувачу засоби для географічного та галузевого аналізу ринку [1, 2].

У процесі роботи проведено огляд сучасних технологій та інструментів веброзробки, на основі чого обґрунтовано вибір стеку технологій (React [16] для фронтенду, Node.js/Express [12, 13] для бекенду, сторонні API Yahoo Finance [21] та World Bank [5] для даних). На етапі проєктування було розроблено архітектуру системи, що включає клієнтську частину з багаторівневою навігацією та серверну частину з REST API для збору і обробки даних [10, 11]. Реалізовано всі компоненти згідно з проєктом: модулі для відображення інтерактивної карти [9], таблиць, графіків [7, 19] та інтерфейсних елементів навігації. Особливу увагу приділено забезпеченню узгодженості даних між фронтендом і бекендом, для чого впроваджено механізми кешування [23] та конвертації валют.

Тестування готової системи підтвердило, що вона відповідає вимогам технічного завдання. По-перше, система успішно інтегрується з реальними джерелами даних: автоматично отримує актуальні біржові дані через API Yahoo Finance [21] та макроекономічні індикатори через API Світового банку [5]. Це забезпечує достовірність та своєчасність інформації, що відображається користувачу. По-друге, розроблений вебінтерфейс виявився зручним і інтуїтивним: реалізовано дворівневу навігацію (за країнами та секторами), яка полегшує доступ до потрібних даних [10, 11]; інтерактивна карта та графіки роблять представлення

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		68

інформації наочним [7, 9, 19]. По-третє, продуктивність застосунку оптимізовано завдяки використанню кешування [23] – при багаторазовому перегляді сторінок відгук системи відбувається майже без затримок, що покращує досвід кінцевого користувача. Застосунок стабільно працює при різних сценаріях експлуатації і коректно обробляє можливі нештатні ситуації (відсутність даних, помилки мережі тощо), що свідчить про його надійність.

Поставлені у вступі роботи цілі досягнуто повністю. Розроблена інформаційна система дозволяє отримувати та аналізувати дані про стан фондового ринку Азії в розрізі країн і галузей, надаючи користувачам зручний інструментарій для оцінки ринкових тенденцій. Виконано усі задачі, окреслені в процесі роботи: проаналізовано предметну область і наявні аналоги, спроектовано архітектуру системи, реалізовано програмний код, інтегровано зовнішні API, проведено тестування функціоналу та продуктивності.

Практичне значення результатів полягає в тому, що розроблений застосунок може бути використаний як основа для більш розгорнутої аналітичної платформи або освітнього інструменту з фінансової грамотності. Система легко масштабується – за потреби можна додати нові країни, розширити перелік компаній чи секторів, або інтегрувати додаткові аналітичні модулі (наприклад, прогнозування на основі історичних даних). Таким чином, робота демонструє можливості сучасних вебтехнологій у вирішенні задач збору та візуалізації фінансових даних, а її результати можуть бути використані для подальших досліджень або практичної реалізації сервісів у фінтех сфері.

На завершення, розробка успішно пройшла всі етапи – від задуму до тестування – і відповідає критеріям, висунутим до дипломного проекту. Отриманий досвід підтверджує доцільність використання вибраного стеку технологій для подібного класу задач та відкриває перспективи для

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ док.ум.	Підпис	Дата		69

удосконалення системи (наприклад, впровадження реального часу оновлення котирувань, розширення аналітичного функціоналу, підтримка більшої кількості регіонів). Поставлені цілі виконано, а результати роботи можуть слугувати основою для подальшого розвитку інформаційно-аналітичних вебсистем у галузі фінансових ринків.

					ІАЛІЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		70

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Мішра А. П. Introduction to Economic Analysis. — Cham : Springer, 2018. — 352 с.
2. Майєр-Шенбергер В., Цук'єр К. Big Data: A Revolution That Will Transform How We Live, Work, and Think. — Boston : Houghton Mifflin Harcourt, 2013. — 304 с.
3. Бриньолфссон Е., Макафі А. Machine, Platform, Crowd: Harnessing Our Digital Future. — New York : W. W. Norton & Company, 2017. — 384 с.
4. International Monetary Fund. International Financial Statistics: database — Washington, DC : IMF, 2025. — <https://data.imf.org> (дата звернення 07.06.2025).
5. World Bank. World Development Indicators: відкриті дані — Washington, DC : World Bank, 2025. — <https://data.worldbank.org> (дата звернення 07.06.2025).
6. Bloomberg L. P. Bloomberg Terminal: Getting Started Guide for Students — New York : Bloomberg, 2020. — 28 с. — <https://data.bloomberglp.com/...> (дата звернення 07.06.2025).
7. Trading Economics: статистичний портал — URL: <https://tradingeconomics.com> (дата звернення 07.06.2025).
8. Chen X., Nordhaus W. Using Luminosity Data as a Proxy for Economic Statistics // Proceedings of the National Academy of Sciences. — 2011. — Vol. 108, No. 21. — P. 8589–8594.
9. Federal Reserve Bank of San Francisco. Asia Economic Policy Conference: proceedings. — San Francisco : FRBSF, 2024. — 186 с.

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		71

10. Acropolium. Modern Web Application Architecture 2025 — URL: <https://acropolium.com/blog/modern-web-application-architecture-2025/> (дата звернення 07.06.2025).
11. MobiDev. Web Application Architecture – Choosing the Best Product — URL: <https://mobidev.biz/blog/web-application-architecture-choosing-best-product> (дата звернення 07.06.2025).
12. Netguru. When, How and Why to Use Node.js for Backend — URL: <https://netguru.com/blog/when-how-why-use-nodejs-backend> (дата звернення 07.06.2025).
13. Express.js — Fast, unopinionated, minimalist web framework for Node.js — URL: <https://expressjs.com> (дата звернення 07.06.2025).
14. Kumar A. (ред.) Flask vs Express JS – Top Differences // GeeksforGeeks, 2024. — URL: <https://geeksforgeeks.org/flask-vs-express-js-top-differences/> (дата звернення 07.06.2025).
15. Amazon Web Services. MongoDB vs MySQL — URL: <https://aws.amazon.com/compare/the-difference-between-mongodb-and-mysql-databases/> (дата звернення 07.06.2025).
16. Crooks M. Angular vs React: Making the Front-End Choice in 2024 Medium, 2024. — URL: <https://medium.com/...> (дата звернення 07.06.2025).
17. DevsData. Angular vs React — URL: <https://devsdata.com/blog/angular-vs-react/> (дата звернення 07.06.2025).
18. Statista. Most Used Web Frameworks Worldwide 2024 — URL: <https://statista.com/statistics/1126786/worldwide-developer-survey-most-used-web-frameworks/> (дата звернення 07.06.2025).
19. Chart.js — URL: <https://en.wikipedia.org/wiki/Chart.js> (дата звернення 07.06.2025).

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		72

20. react-simple-maps: npm-пакунок — URL: <https://npmjs.com/package/react-simple-maps> (дата звернення 07.06.2025).
21. yahoo-finance2: npm-пакунок [Електронний ресурс]. — URL: <https://npmjs.com/package/yahoo-finance2> (дата звернення 07.06.2025).
22. Mi T. How to Fetch Stock Market Data Using Yahoo Finance API in Node.js [Електронний ресурс] // OMI.me Blog, 2024. — URL: <https://omi.me/blog/how-to-fetch-stock-market-data-using-yahoo-finance-api-in-nodejs/> (дата звернення 07.06.2025).
23. Axios Documentation [Електронний ресурс]. — URL: <https://axios-http.com/docs/intro> (дата звернення 07.06.2025).
24. Jena A. A Beginner’s Guide to Using Axios with Promises in JavaScript [Електронний ресурс] // Medium, 2023. — URL: <https://medium.com/@ashishkumarjena/...> (дата звернення 07.06.2025).
25. concurrently: npm-пакунок [Електронний ресурс]. — URL: <https://npmjs.com/package/concurrently> (дата звернення 07.06.2025).
26. GeeksforGeeks. Restart Node.js Server Automatically with Nodemon [Електронний ресурс]. — URL: <https://geeksforgeeks.org/restart-nodejs-server-automatically-with-nodemon/> (дата звернення 07.06.2025).
27. Tailwind CSS: офіційна документація [Електронний ресурс]. — URL: <https://tailwindcss.com> (дата звернення 07.06.2025).
28. Tailwind CSS vs Bootstrap [Електронний ресурс] // Agicent Blog, 2023. — URL: <https://agicent.com/blog/tailwind-css-vs-bootstrap/> (дата звернення 07.06.2025).
29. Advantages of Tailwind CSS over other front-end frameworks [Електронний ресурс] // Quora Q&A. — URL: <https://quora.com/...> (дата звернення 07.06.2025).

30. AlgoTrading 101. Yahoo Finance API Guide [Електронний ресурс]. — URL: <https://algotrading101.com/learn/yahoo-finance-api/> (дата звернення 07.06.2025).
31. World Bank. Indicators API Documentation [Електронний ресурс]. — URL: [https://datahelpdesk.worldbank.org/...](https://datahelpdesk.worldbank.org/) (дата звернення 07.06.2025).
32. Cachefly. Comprehensive Overview of Caching [Електронний ресурс]. — URL: <https://cachefly.com/blog/comprehensive-overview-of-caching/> (дата звернення 07.06.2025). Next.JS vs React: The Ultimate Comparison Guide (2025). Taction Software. URL: <https://www.tactionsoft.com/guide/next-js-vs-react-comparison/> (дата звернення: 10.05.2025).
33. daleogont/Asian-Economy-Analysis GitHub. URL: <https://github.com/daleogont/Asian-Economy-Analysis> (дата звернення: 29.05.2025)

					ІАЛЦ.467200.003 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		74

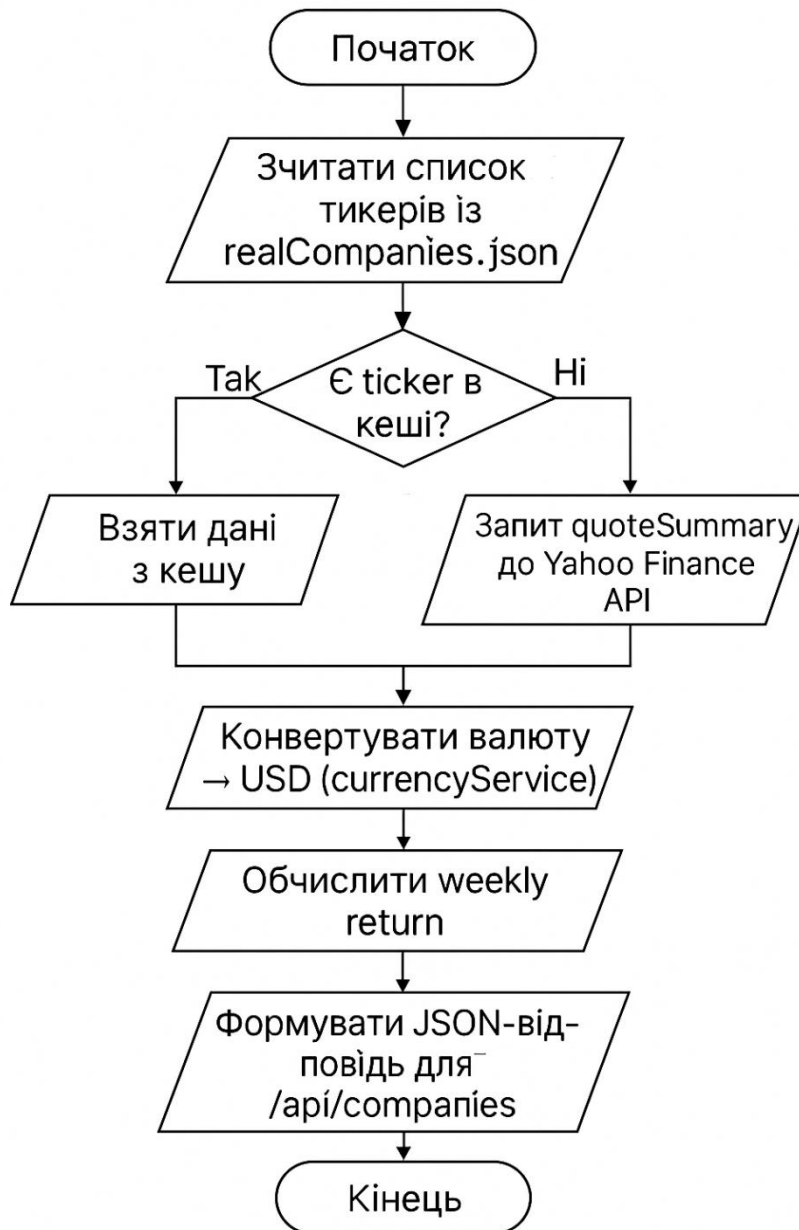
# ДОДАТОК 1

Веб-застосунок для обліку та аналізу особистих фінансів

Алгоритм формування біржових даних

Аркушів 1

Київ 2025



					ІАЛЦ.467200.004 Д1			
Зм.	Аркуш	№ докум.	Підпис	Дата				
Розробив	Гонтар Д. С.				Веб-застосунок для економічного аналізу азійського регіону Алгоритм формування біржових даних	Літ.	Аркуш	Аркушів
Перевірів	Кулаков Ю. О.						1	1
Н. Контр.	Нікольский С.С.					КПІ ім. Ігоря Сікорського, ФІОТ, ІО-15		
Затв.								

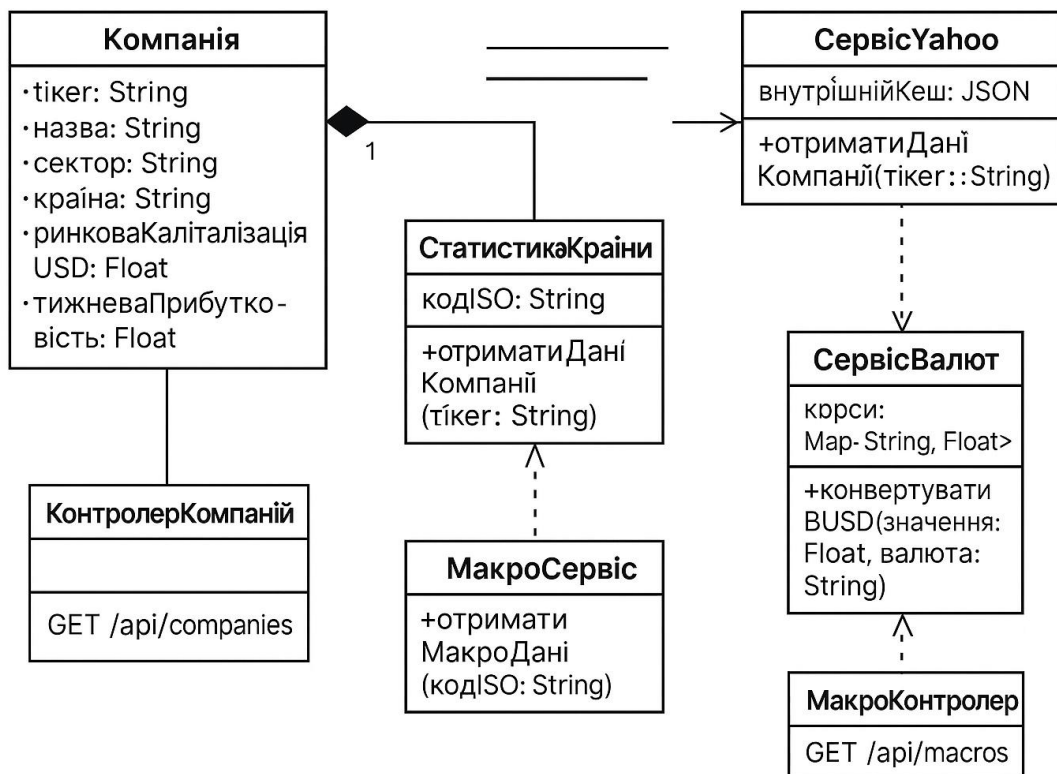
## ДОДАТОК 2

Веб-застосунок для економічного аналізу азійського регіону

Діаграма класів

Аркушів 1

Київ 2025



					ІАЛЦ.467200.005 Д2			
Зм.	Аркуш	№ докум.	Підпис	Дата				
Розробив	Гонтар Д. С.				Веб-застосунок для економічного аналізу азійського регіону	Літ.	Аркуш	Аркушів
Перевірів	Кулаков Ю. О.						1	1
Н. Контр.	Нікольский С.С.					КПІ ім. Ігоря Сікорського, ФІОТ, ІО-15		
Затв.								
					Діаграма класів			

## ДОДАТОК 3

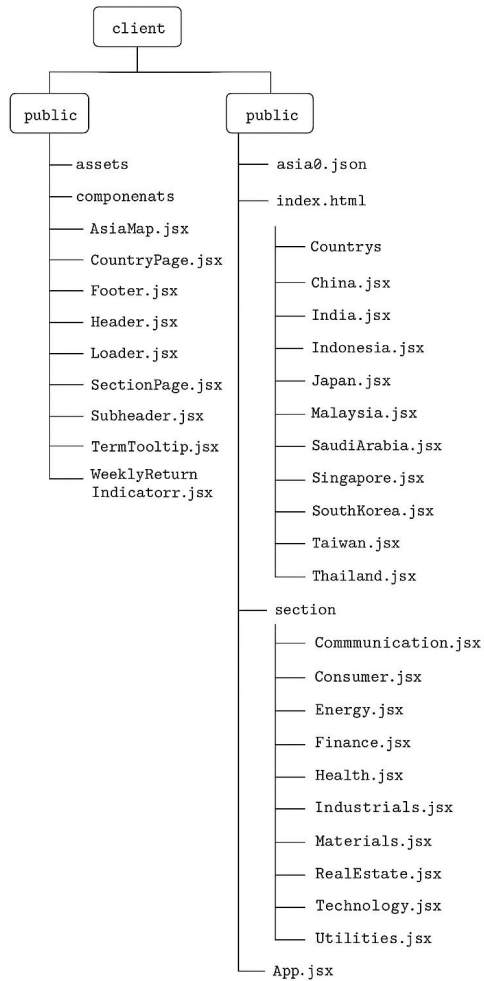
Веб-застосунок для економічного аналізу азійського регіону

Структурна модель системи

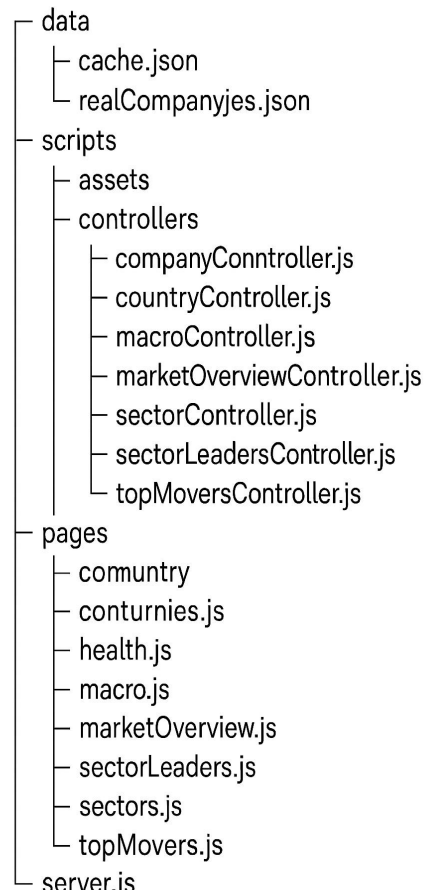
Аркушів 1

Київ 2025

## Frontend-схема



## Backend-схема



					ІАЛЦ.467200.006 ДЗ							
Зм.	Аркуш	№ докум.	Підпис	Дата	Веб-застосунок для економічного аналізу азійського регіону  Структурна модель системи			Літ.	Аркуш	Аркушів		
Розробив	Гонтар Д. С.									1	1	
Перевірив	Кулаков Ю. О.											
Н. Контр.	Нікольский С.С.							КПІ ім. Ігоря Сікорського, ФІОТ, ІО-15				
Затв.												

## ДОДАТОК 4

Веб-застосунок для економічного аналізу азійського регіону

Текст програмного коду

Аркушів 9

Київ 2025

```

server/server.js
const express = require("express");
const cors = require("cors");
const path = require("path");
const fs = require("fs");

const cachePath = path.join(__dirname, "../data/cache.json");

const companiesRouter = require("./routes/companies");
const sectorsRouter = require("./routes/sectors");
const countriesRouter = require("./routes/countries");
const macroRouter = require("./routes/macro");
const healthRouter = require("./routes/health");

const topMoversRouter = require("./routes/topMovers");
const sectorLeadersRouter = require("./routes/sectorLeaders");
const marketOverviewRouter = require("./routes/marketOverview");

const app = express();
const PORT = require("../config").PORT || 4000;

app.use(cors());
app.use(express.json());

app.use("/api/companies", companiesRouter);
app.use("/api/sectors", sectorsRouter);
app.use("/api/countries", countriesRouter);
app.use("/api/macros", macroRouter);
app.use("/api/top-movers", topMoversRouter);
app.use("/api/sector-leaders", sectorLeadersRouter);
app.use("/api/market-overview", marketOverviewRouter);

app.get("/", (req, res) => {
  res.send("Asian Economy Analysis API is running.");
});
app.use("/api/health", healthRouter);

app.use((err, req, res, next) => {
  console.error("Server error:", err);
  res.status(500).json({ error: "Server error" });
});

app.listen(PORT, () => {
  console.log(`Сервер запущено на порту ${PORT}`);
});

process.on("SIGINT", () => {
  console.log("\nОтримано SIGINT — очищуємо cache.json...");
  try {
    if (fs.existsSync(cachePath)) {
      fs.writeFileSync(cachePath, "{}");
      console.log("Файл cache.json очищено ({}).");
    }
  }
}

```

					ІАЛЦ.467200.007 Д4			
Зм.	Аркуш	№ докум.	Підпис	Дата	Веб-застосунок для економічного аналізу азійського регіону	Літ.	Аркуш	Аркушів
Розробив	Гонтар Д. С.						1	9
Перевірів	Кулаков Ю. О.							
Н. Контр.	Нікольский С.С.					КПІ ім. Ігоря Сікорського, ФІОТ, ІО-15		
Затв.								
					Текст програмного коду			

```

    } catch (err) {
      console.error(" Не вдалося очистити cache.json:", err.message);
    }
    process.exit();
  });

server/controllers/companyController.js
const path = require("path");
const fs = require("fs");
const { fetchCompanyData } = require("../utils/yahooService");

const companiesPath = path.join(__dirname, "../data/realCompanies.json");

const getCompanies = async (req, res) => {
  try {
    const rawData = fs.readFileSync(companiesPath, "utf-8");
    const companies = JSON.parse(rawData);

    const results = await Promise.all(
      companies.map(async (company) => {
        try {
          const data = await fetchCompanyData(company.ticker);

          if (
            !data ||
            typeof data.marketCap !== "number" ||
            typeof data.stockPrice !== "number" ||
            typeof data.previousClose !== "number"
          ) {
            console.warn(` Недостатньо даних для ${company.ticker}`);
            return null;
          }

          const weeklyReturn =
            ((data.stockPrice - data.previousClose) / data.previousClose) * 100;

          return {
            name: company.name,
            ticker: company.ticker,
            country: company.country,
            sector: company.sector,
            marketCap: data.marketCap,
            stockPrice: data.stockPrice,
            weeklyReturn: Number(weeklyReturn.toFixed(2)),
          };
        } catch (err) {
          console.error(` Помилка при запиті до Yahoo для ${company.ticker}:`, err.message);
          return null;
        }
      })
    );
  }

  const unique = {};
  results.filter(Boolean).forEach((company) => {
    const key = `${company.country}${company.sector}`;
    if (!unique[key] || company.stockPrice > unique[key].stockPrice) {
      unique[key] = company;
    }
  });
});

```

					ІАЛІЦ.467200.009 Д6	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		2

```

const sorted = Object.values(unique).sort((a, b) => {
  if (a.country !== b.country) return a.country.localeCompare(b.country);
  return a.sector.localeCompare(b.sector);
});

res.json(sorted);
} catch (error) {
  console.error("❖❖ Фатальна помилка в getCompanies:", error.message);
  res.status(500).json({ error: "Не вдалося отримати дані про компанії" });
}
};

```

```
module.exports = { getCompanies };
```

```
server/routes/companies.js
```

```

const express = require("express");
const router = express.Router();
const { getCompanies } = require("../controllers/companyController");

```

```
router.get("/", getCompanies);
```

```
module.exports = router;
```

```
scripts/fetchAllToCache.js
```

```

const yahooFinance = require("yahoo-finance2").default;
const fs = require("fs");
const path = require("path");
const { convertToUSD } = require("../server/utills/currencyService");
const companies = require("../data/realCompanies.json");

```

```
const cachePath = path.join(__dirname, "../data/cache.json");
```

```
(async () => {
  const cache = {};

```

```

  for (const company of companies) {
    const ticker = company.ticker;
    try {
      const quote = await yahooFinance.quoteSummary(ticker, {
        modules: ["price", "summaryDetail", "financialData"],
      });

```

```

      const price = quote?.price || {};
      const currency = price.currency || "USD";

```

```

      cache[ticker] = {
        ticker,
        stockPrice: convertToUSD(price.regularMarketPrice ?? 0, currency),
        marketCap: convertToUSD(price.marketCap ?? 0, currency),
        previousClose: convertToUSD(price.regularMarketPreviousClose ?? 0, currency),
        currency,
      };

```

```

      console.log(`[+] ${ticker} — OK`);
    } catch {
      cache[ticker] = null;
      console.log(`[x] ${ticker} — error`);
    }
  }
}

```

					ІАЛІЦ.467200.009 Д6	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		3

```
fs.writeFileSync(cachePath, JSON.stringify(cache, null, 2));
console.log("Готово! Дані збережено у /data/cache.json");
process.exit(0);
})();
```

```
server/utills/currencyService.js
```

```
const currencyToUSD = {
  JPY: 0.006934,
  CNY: 0.1391,
  HKD: 0.1274,
  INR: 0.01164,
  SGD: 0.7755,
  KRW: 0.0007,
  IDR: 0.0000612,
  MYR: 0.2353,
  TWD: 0.0334,
  SAR: 0.266667,
  THB: 0.0308,
  USD: 1
};

function convertToUSD(value, currency) {
  const rate = currencyToUSD[currency];
  if (rate === null) {
    console.warn(`[currencyService] Невідомий курс для валюти "${currency}"`);
    return value;
  }
  return value * rate;
}

module.exports = {
  convertToUSD
};
```

```
server/utills/yahooService.js
```

```
const yahooFinance = require("yahoo-finance2").default;
const fs = require("fs");
const path = require("path");
const { convertToUSD } = require("../currencyService");

const cachePath = path.resolve(__dirname, "../data/cache.json");

let cache = {};
if (fs.existsSync(cachePath)) {
  try {
    cache = JSON.parse(fs.readFileSync(cachePath, "utf-8"));
  } catch {
    cache = {};
  }
}

const fetchCompanyData = async (ticker) => {
  if (cache[ticker]) {
    return cache[ticker];
  }
  try {
    const quote = await yahooFinance.quoteSummary(ticker, {
      modules: ["price", "summaryDetail", "financialData"],
    });
  }
}
```

					ІАЛІЦ.467200.009 Д6	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		4

```

});

if (!quote || !quote.price) return null;

const price = quote.price;
const finance = quote.financialData || {};
const currency = price.currency || "USD";

if (
  typeof price.regularMarketPrice !== "number" ||
  typeof price.regularMarketPreviousClose !== "number" ||
  typeof price.marketCap !== "number"
) return null;

const stockPriceUSD = convertToUSD(price.regularMarketPrice, currency);
const marketCapUSD = convertToUSD(price.marketCap, currency);

const data = {
  ticker,
  stockPrice: stockPriceUSD,
  marketCap: marketCapUSD,
  peRatio: finance.trailingPE ?? null,
  previousClose: convertToUSD(price.regularMarketPreviousClose, currency),
  currency,
};

cache[ticker] = data;
fs.writeFileSync(cachePath, JSON.stringify(cache, null, 2));

return data;
} catch (error) {
  console.error(` Yahoo error for ${ticker}:`, error.message);
  return null;
}
};

module.exports = { fetchCompanyData };

```

```

client/src/app.jsx
import React, { useState, useEffect } from "react";
import { Routes, Route } from "react-router-dom";
import axios from "axios";

import Header from "./components/Header";
import Subheader from "./components/Subheader";
import Footer from "./components/Footer";
import Loader from "./components/Loader";

import Home from "./pages/Home";
import Research from "./pages/Research";

import Sections from "./pages/Sections";
import Finance from "./pages/section/Finance";
import Technology from "./pages/section/Technology";
import Energy from "./pages/section/Energy";
import Health from "./pages/section/Health";
import RealEstate from "./pages/section/RealEstate";
import Utilities from "./pages/section/Utilities";
import Industrials from "./pages/section/Industrials";
import Consumer from "./pages/section/Consumer";

```

					ІАЛІЦ.467200.009 Д6	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		5

```

import Materials from "./pages/section/Materials";
import Communication from "./pages/section/Communication";

import Countries from "./pages/Countries";
import Japan from "./pages/country/Japan";
import China from "./pages/country/China";
import India from "./pages/country/India";
import Thailand from "./pages/country/Thailand";
import Singapore from "./pages/country/Singapore";
import Indonesia from "./pages/country/Indonesia";
import Malaysia from "./pages/country/Malaysia";
import Taiwan from "./pages/country/Taiwan";
import SouthKorea from "./pages/country/SouthKorea";
import SaudiArabia from "./pages/country/SaudiArabia";

const App = () => {
  const [isServerReady, setIsServerReady] = useState(false);

  useEffect(() => {
    axios
      .get("/api/health")
      .then(() => {
        setIsServerReady(true);
      })
      .catch((err) => {
        console.error("Помилка health-check:", err.message);
        setTimeout(() => {
          axios
            .get("/api/health")
            .then(() => setIsServerReady(true))
            .catch(() => console.error("Знову не готово."));
        }, 2000);
      });
  }, []);

  if (!isServerReady) {
    return <Loader />;
  }

  return (
    <div className="pt-32 pb-20">
      <Header />
      <Subheader />

      <Routes>
        <Route path="/" element={<Home />} />
        <Route path="/research" element={<Research />} />

        <Route path="/sections" element={<Sections />} />
        <Route path="/section/finance" element={<Finance />} />
        <Route path="/section/technology" element={<Technology />} />
        <Route path="/section/energy" element={<Energy />} />
        <Route path="/section/health" element={<Health />} />
        <Route path="/section/realstate" element={<RealEstate />} />
        <Route path="/section/utilities" element={<Utilities />} />
        <Route path="/section/industrials" element={<Industrials />} />
        <Route path="/section/consumer" element={<Consumer />} />
        <Route path="/section/materials" element={<Materials />} />
        <Route path="/section/communication" element={<Communication />} />

        <Route path="/countries" element={<Countries />} />

```

					ІАЛІЦ.467200.009 Д6	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		6

```

    <Route path="/country/japan" element={<Japan />} />
    <Route path="/country/china" element={<China />} />
    <Route path="/country/india" element={<India />} />
    <Route path="/country/thailand" element={<Thailand />} />
    <Route path="/country/singapore" element={<Singapore />} />
    <Route path="/country/indonesia" element={<Indonesia />} />
    <Route path="/country/malaysia" element={<Malaysia />} />
    <Route path="/country/taiwan" element={<Taiwan />} />
    <Route path="/country/southkorea" element={<SouthKorea />} />
    <Route path="/country/saudiArabia" element={<SaudiArabia />} />
  </Routes>

  <Footer />
</div>
);
};

export default App;

client/src/pages/home.jsx

import React, { useEffect, useState } from "react";
import axios from "axios";
import AsiaMap from "../components/AsiaMap";

const Home = () => {
  const [companies, setCompanies] = useState([]);
  const [sectors, setSectors] = useState([]);
  const [countries, setCountries] = useState([]);
  const [topMovers, setTopMovers] = useState([]);
  const [sectorLeaders, setSectorLeaders] = useState([]);

  useEffect(() => {
    axios.get("/api/companies").then(res => setCompanies(res.data));
    axios.get("/api/sectors").then(res => setSectors(res.data));
    axios.get("/api/countries").then(res => setCountries(res.data));
    axios.get("/api/top-movers").then(res => setTopMovers(res.data));
    axios.get("/api/sector-leaders").then(res => setSectorLeaders(res.data));
  }, []);

  const totalMarketCap = companies.reduce((sum, c) => sum + (c.marketCap || 0), 0);
  const avgWeeklyReturn = companies.length
    ? companies.reduce((sum, c) => sum + (c.weeklyReturn || 0), 0) / companies.length
    : 0;

  const topSector = sectors.reduce((max, s) => (s.marketWeight > (max?.marketWeight || 0) ? s : max),
    null);
  const topCountry = countries.reduce((max, c) => (c.totalCap > (max?.totalCap || 0) ? c : max), null);

  return (
    <div className="px-6 py-8">
      <h1 className="text-3xl font-bold mb-6 text-center">Asia Market Dashboard</h1>

      <div className="grid grid-cols-1 md:grid-cols-4 gap-6 mb-8">
        <div className="bg-white rounded shadow p-6 text-center">
          <div className="text-gray-500">Total Market Cap</div>
          <div className="text-2xl font-bold">{(totalMarketCap / 1e12).toFixed(2)} трлн $</div>
        </div>
        <div className="bg-white rounded shadow p-6 text-center">
          <div className="text-gray-500">Avg Weekly Return</div>
        </div>
      </div>
    </div>
  );
};

```

					ІАЛІЦ.467200.009 Д6	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		7

```

<div className={`text-2xl font-bold ${avgWeeklyReturn > 0 ? "text-green-600" : "text-red-600"}`}>
  {avgWeeklyReturn.toFixed(2)}%
</div>
</div>
<div className="bg-white rounded shadow p-6 text-center">
  <div className="text-gray-500">Top Sector</div>
  <div className="text-xl font-bold">{topSector?.name || "-"}</div>
  <div className="text-gray-500">{topSector ? topSector.marketWeight.toFixed(1) + "%" : "-"}</div>
</div>
<div className="bg-white rounded shadow p-6 text-center">
  <div className="text-gray-500">Top Country</div>
  <div className="text-xl font-bold">{topCountry?.country || "-"}</div>
  <div className="text-gray-500">{topCountry ? (topCountry.totalCap / 1e9).toFixed(1) + " млрд"}</div>
</div>

<AsiaMap />

<div className="grid grid-cols-1 md:grid-cols-2 gap-8 mt-10">
  <div>
    <h2 className="text-lg font-semibold mb-2">Top Movers (Weekly Return)</h2>
    <table className="min-w-full bg-white border border-gray-200 rounded-lg shadow text-sm">
      <thead className="bg-gray-100">
        <tr>
          <th className="px-3 py-2 text-left">Company</th>
          <th className="px-3 py-2 text-left">Country</th>
          <th className="px-3 py-2 text-right">Weekly Return (%)</th>
        </tr>
      </thead>
      <tbody>
        {topMovers.slice(0, 5).map((c, idx) => (
          <tr key={c.ticker} className={idx % 2 === 0 ? "bg-white" : "bg-gray-50"}>
            <td className="px-3 py-2">{c.name}</td>
            <td className="px-3 py-2">{c.country}</td>
            <td className={`px-3 py-2 text-right ${c.weeklyReturn > 0 ? "text-green-600" : "text-red-600"}`}>
              {c.weeklyReturn.toFixed(2)}
            </td>
          </tr>
        ))}
      </tbody>
    </table>
  </div>
  <div>
    <h2 className="text-lg font-semibold mb-2">Sector Leaders</h2>
    <table className="min-w-full bg-white border border-gray-200 rounded-lg shadow text-sm">
      <thead className="bg-gray-100">
        <tr>
          <th className="px-3 py-2 text-left">Sector</th>
          <th className="px-3 py-2 text-left">Company</th>
          <th className="px-3 py-2 text-right">Market Cap ($)</th>
        </tr>
      </thead>
      <tbody>
        {sectorLeaders.map((c, idx) => (
          <tr key={c.ticker} className={idx % 2 === 0 ? "bg-white" : "bg-gray-50"}>
            <td className="px-3 py-2">{c.sector}</td>
            <td className="px-3 py-2">{c.name}</td>
            <td className="px-3 py-2 text-right">{(c.marketCap / 1e9).toFixed(1)} млрд</td>
          </tr>
        ))}
      </tbody>
    </table>
  </div>
</div>

```

					ІАЛІЦ.467200.009 Д6	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		8

